



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Académico Profesional de Ingeniería de Sistemas

Propuesta del sistema de pago y gestión de cobranza vía web de la Facultad de Ingeniería de Sistemas de UNMSM

TESINA

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

William Rafael CUBAS MACAZANA

ASESOR

Hugo VEGA HUERTA

Lima, Perú

2010



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Cubas, W. (2010). *Propuesta del sistema de pago y gestión de cobranza vía web de la Facultad de Ingeniería de Sistemas de UNMSM*. Tesina para optar el título de Ingeniero de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

FICHA CATALOGRÁFICA

CUBAS MACAZANA WILLIAM RAFAEL

**PROPUESTA del SISTEMA DE PAGOS y GESTIÓN de
COBRANZAS VÍA WEB de la FACULTAD de INGENIERÍA
de SISTEMAS e INFORMÁTICA de la UNMSM.**

(LIMA) 2010.

**XVII, 100p., 29.7 cm (UNMSM INGENIERO, SISTEMAS,
2010).**

**Tesis, Universidad Nacional Mayor de San Marcos, Facultad de
Ingeniería de Sistema e Informática.**

DEDICATORIA

A mi Madre sólo a ella.

AGRADECIMIENTOS

A mis Padres, Profesores y amigos
que siempre estuvieron cerca mío.

RESUMEN

PROPUESTA DEL SISTEMA DE PAGO Y GESTIÓN DE COBRANZA

VÍA WEB DE LA FACULTAD DE INGENIERÍA DE SISTEMAS DE UNMSM

Bach. Cubas Macazana, William Rafael

February – 2010

Asesor : Vega Huertas, Hugo

Grado a obtener : Ingeniero de Sistemas

Para la solución de problema del trabajo se revisaron las necesidades requeridas por el personal encargado de las cobranzas, pagos y deudas, dando como resultado el planteamiento de un prototipo de solución a los problemas existentes a través de un Sistema de Pago y Gestión de Cobranza Vía Web de la Facultad de Ingeniería de Sistemas de UNMSM desarrollado en el presente trabajo de Titulación.

Dicho proyecto consiste en la construcción del software que controlará a los estudiantes que mantengan alguna deuda a la FISI y luego a otros usuarios como los egresados, bachilleres, etc., el cual permitirá resolver la cobertura web sobre sus reportes de deudas, basada en la necesidad de incrementar la eficiencia en los pagos.

PALABRAS CLAVES:

- Prototipo
- Software
- Web

ABSTRACT

PROPOSAL OF SYSTEM OF PAYMENT AND CHARGES MANAGEMENT WAY WEB OF FACULTY OF SYSTEMS ENGINEERING OF UNMSM

Bach. Cubas Macazana, William Rafael

January - 2010

Adviser : Vega Huertas, Hugo

Degree to obtain : Systems Engineer

To solve the problem of work required needs were reviewed by the staff responsible for Charges, payments and debts, giving as resulted in a prototype approach for solving the existing problems through a Payment System for Charges and Management Way Website of the Faculty of Engineering of UNMSM systems developed in this work Degree.

This project involves the construction of software that controls students to maintain a debt to the FISI and then to other users such as graduates, bachillers etc., Which will resolve on its web coverage of debt reports, based on the need for increased efficiency in payment.

.KEY WORDS:

- Prototype
- Software
- Web

ÍNDICE

TEMA	PÁG
INTRODUCCIÓN	1
ANTECEDENTES.....	2
CAPÍTULO I	3
1.1 EL PROBLEMA.....	3
1.1.1 PROBLEMA PRINCIPAL	3
1.1.2 PROBLEMA SECUNDARIO.....	3
1.2 DELIMITACIÓN ESPACIAL	3
1.3 JUSTIFICACIÓN	4
1.3.1 EL ASPECTO TEÓRICO.....	4
1.4 OBJETIVOS	4
1.4.1 OBJETIVOS GENERALES.....	4
1.4.2 OBJETIVOS ESPECÍFICOS.....	4
1.5 ORGANIZACIÓN DE LA TESINA	5
CAPÍTULO II: MARCO TEÓRICO	6
2.1 METODOLOGÍA RUP	6
2.1.1 PROCESO DIRIGIDO POR CASOS DE USO	6
2.1.2 PROCESO CENTRADO EN LA ARQUITECTURA	7
2.1.3 PROCESO ITERATIVO E INCREMENTAL	7
2.1.4 ESTRUCTURA DE RUP	7
2.1.5 ESTRUCTURA DINÁMICA DEL PROCESO	8
2.1.5.1 Inicio	9
2.1.5.2 Elaboración	9
2.1.5.3 Construcción	9

2.1.5.4 Transición	10
2.2 METODOLOGÍA SCRUM	10
2.2.1 DESARROLLO INCREMENTAL	10
2.2.1.1 Desarrollo evolutivo.....	10
2.2.1.2 Auto-organización.....	11
2.2.1.3 Colaboración	11
2.2.2 VISIÓN GENERAL DEL PROCESO.....	11
2.2.2.1 Las Reuniones	12
2.2.2.2 Los Elementos.....	12
2.2.2.3 Los Roles o Responsabilidades.....	13
2.2.3 SCRUM MANAGEMENT	13
2.3 LA PROGRAMACIÓN EXTREMA.....	14
2.3.1 Actividades de Xp.....	14
2.3.1.1 Codificar.....	14
2.3.1.2 Hacer pruebas.....	14
2.3.1.3 Escuchar	15
2.3.1.4 Diseñar	15
2.3.1.5 Prácticas Básicas de XP	15
2.3.2 CICLO DE VIDA XP	17
2.3.2.1 Fase de la exploración.....	18
2.3.2.2 Fase del planeamiento.....	18
2.3.2.3 Fase de producción.....	19
2.3.2.4 Fase de mantenimiento.....	19
2.3.2.5 Fase de muerte	19
2.4 MARCO INDIVIDUAL	19

2.5 MARCO LEGAL.....	19
2.6 ESQUEMA DE LA METODOLOGÍA DEL SISTEMA	20
CAPÍTULO III: ESTADO DEL ARTE.....	21
3.1 BENCHMARKING ENTRE METODOLOGÍAS EXISTENTES.....	21
3.1.1 ANÁLISIS DE LOS CRITERIOS DE EVALUACIÓN	21
3.1.1.1 Sistema como algo cambiante.....	21
3.1.1.2 Colaboración entre los miembros del equipo.....	21
3.1.1.3 Características Metodológicas (CM)	21
3.2 SELECCIÓN DE LA HERRAMIENTA.....	24
3.2.1 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE	24
3.2.1.1 PHP (HyperText PreProcesador)	24
3.2.1.2 J2EE (Java 2 Enterprise Edition)	26
3.2.1.3 ASP (Active Server Pages)	28
3.2.1.4 .NET.....	29
3.3 BENCHMARKING ENTRE LAS HERRAMIENTAS DE DESARROLLO	29
3.3.1 ANÁLISIS DE LOS CRITERIOS DE EVALUACIÓN	30
3.3.2 SELECCION DE LA BASE DE DATOS	32
3.3.2.1 Microsost SQL server 2005 Express.....	32
3.3.2.2 PostGreSQL	33
3.3.2.3 MySQL	34
3.3.3 COMPARACION ENTRE BASE DE DATOS	36
3.3.4 RESUMEN DE LAS COMPARACIONES	37
3.3.5 ESQUEMA DEL DESARROLLO DEL SISTEMA	38

CAPÍTULO IV: ANÁLISIS Y DISEÑO DEL SISTEMA	39
4.1 ESQUEMA INPUT DEL SISTEMA.....	39
4.2 VISIÓN	40
4.2.1 INTRODUCCIÓN	40
4.2.1.1 Propósito	40
4.2.1.2 Alcance	40
4.2.1.3 Definiciones, Acrónimos, y Abreviaciones	40
4.2.2 POSICIONAMIENTO.....	40
4.2.2.1 Oportunidad de Negocio	40
4.2.2.2 Planteamiento del problema.....	41
4.2.2.3 Planteamiento de la posición del Servicio	42
4.2.3 DESCRIPCIÓN DE STAKEHOLDERS Y USUARIOS.....	42
4.2.3.1 Resumen de Stakeholders	43
4.2.3.2 Resumen de Usuarios.....	43
4.2.3.3 Ambiente de Usuario	44
4.2.3.4 Perfil de los Stakeholders.....	44
4.2.3.5 Perfiles de Usuario.	45
4.2.3.6 Representante Necesidades de los Stakeholders y los Usuarios	46
4.2.3.7 Alternativas y Competencia.....	46
4.2.4 DESCRIPCIÓN GLOBAL DEL SERVICIO	46
4.2.4.1 Perspectiva del Servicio	46
4.2.4.2 Resumen de capacidades.....	47
4.2.4.3 Suposiciones y dependencias	47
4.2.4.5 Costo y precio	47

4.3 ESPECIFICACIONES SUPLEMENTARIAS	47
4.3.1 INTRODUCCIÓN	47
4.3.1.1 Propósito	47
4.3.1.2 Alcance	48
4.3.1.3 Definiciones, Acrónimos, y Abreviaciones	48
4.3.2 USABILIDAD	48
4.3.2.1 Facilidad de uso	48
4.3.2.2 Capacitación.....	48
4.3.2.3 Interfaces de Usuario	48
4.3.2.4 Disponibilidad.....	48
4.3.2.5 Desempeño.....	49
4.3.2.5.1 Tiempo de respuesta.....	49
4.3.2.5.2 Carga de Trabajo.....	49
4.3.2.5.3 Recursos.....	49
4.4 ANÁLISIS	50
4.4.1 MODELOS DE CASOS DE USO GENERAL DEL SISTEMA	50
4.4.2 LISTA DE CASOS DE USO.....	51
4.4.2.1 Caso de uso Ingreso de Deudas de Alumnos	51
4.4.2.2 Caso de uso Realizar Pago de Alumno	52
4.4.2.3 Caso de uso Modificar Información de Deudas.....	53
4.4.2.4 Caso de uso Realizar Pago Fraccionado de Alumnos.....	54
4.4.2.5 Caso de uso Reporte de Deuda	55
4.4.2.6 Caso de uso Reporte de Cobranzas	56
4.4.2.7 Caso de uso Administración de Usuarios	57
4.4.2.8 Caso de uso Ingreso de Usuarios	57

4.4.2.9 Caso de uso Modificar datos de Usuarios.....	58
4.4.2.10 Caso de uso Eliminar Usuarios	59
4.4.2.11 Caso de uso del Sistema.....	60
4.4.3 ESPECIFICACIÓN DE CASOS DE USO	61
4.4.3.1 Ingreso de Deudas de Alumnos	61
4.4.3.2 Realizar Pago de Alumno	62
4.4.3.3 Realizar Pago Fraccionado de Alumnos	63
4.4.3.4 Reporte de Cobranzas	64
4.4.3.5 Reporte de Deudas	65
4.4.3.6 Modificar Información de Deudas de Alumnos.....	66
4.4.3.7 Administrar Usuarios	67
4.5 DISEÑO	68
4.5.1 ARQUITECTURA DEL SISTEMA.....	68
4.5.1.1 Introducción	68
4.5.1.1.1 Propósito	68
4.5.1.1.2 Alcance.....	68
4.5.1.1.3 Definiciones, Acrónimos y Abreviaturas.....	68
4.5.1.1.4 Referencias.....	68
4.5.1.2 Representación de la Arquitectura	68
4.5.1.2.1 Vista de Casos de Uso.....	68
4.5.1.2.2 Vista Lógica	69
4.5.1.2.3 Vista de Despliegue	69
4.5.1.2.4 Vista de Implementación	69
4.5.1.3 Vista de Casos de Uso.....	69
4.5.1.3.1 Modelos de Actores	70

4.5.1.4 Vista Lógica	70
4.5.1.4.1 Vista General.....	70
4.5.1.5 Vista de Despliegue	74
4.5.1.5.1 Servidor Web	74
4.5.1.5.2 Servidor Base de Datos	74
4.5.1.5.3 Estaciones Cliente	74
4.5.1.6 Vista de Implementación	75
4.5.1.6.1 Relación entre Capas y Paquetes	75
4.5.1.6.2 Capas	75
4.5.1.7 Vista de Componentes	76
4.5.1.8 Vista de Colaboración.....	76
4.5.1.8.1 Realizar Pago de Alumnos.....	76
4.5.1.8.2 Realizar Pago Fraccionado de Alumnos	77
4.5.2 REALIZACIÓN DE CASOS DE USO	77
4.5.2.1 Realizar Pago de Alumno	77
4.5.2.1.1 Diagrama de Estados.....	77
4.5.2.1.2 Diagrama de Secuencias	78
4.5.2.2 Realizar Pago Fraccionado de Alumnos	79
4.5.2.2.1 Diagrama de Estados.....	79
4.5.2.2.2 Diagrama de Secuencias	80
4.5.3 DISEÑO DE LA BASE DE DATOS	81
4.5.4 CONFIGURACIÓN DEL SISTEMA.....	82
4.5.4.1 Instalación de la conexión ODBC.....	82
4.5.4.2 Instalar servidor independiente de BD.....	82

4.5.5 PROTOTIPO DEL SISTEMA.....	83
4.5.5.1 Ingreso de Login	83
4.5.5.2 Menú Principal.....	83
4.5.5.3 Ingresar Deudas de Alumnos	84
4.5.5.4 Cobranza del Pago del Alumno	85
4.5.5.5 Modificación de las Deudas del Alumno.....	86
4.5.5.6 Pago Fraccionado de las Deudas del Alumno.....	87
4.5.5.7 Reporte de Alumnos con Deudas.....	88
4.5.5.8 Reporte de Cobranzas	88
4.5.5.9 Listado de Cobranzas General	89
4.5.5.10 Reporte de Pagos de Morosos.....	89
4.5.5.11 Listado de Deudas.....	90
4.5.5.12 Reporte de Deudas Acumuladas	90
4.5.5.13 Administración de Usuarios.....	91
4.5.5.14 Administración de Alumno.....	92
4.5.5.15 Administración de Departamento	93
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	94
5.1 CONCLUSIONES	94
5.2 RECOMENDACIONES.....	94
GLOSARIO DE TÉRMINOS.....	95
BIBLIOGRAFÍA	99

ÍNDICES DE FIGURAS:

FIGURA	PÁG
Figura 2.1: Los Casos de Uso integran el trabajo	6
Figura 2.2: Estructura de RUP	8
Figura 2.3: Iteraciones del Scrum	11
Figura 2.4: Reuniones del Scrum.....	12
Figura 2.5: Scrum Management.....	13
Figura 2.6: Ciclo de vida de la Programación.....	18
Figura 4.1: Modelo de Casos de Uso General	50
Figura 4.2: Modelo de Caso de Uso Ingreso de Deudas de Alumnos	51
Figura 4.3: Modelo de Caso de Uso Realizar Pago de Alumno	52
Figura 4.4: Modelo de Caso de Uso Modificar Información de Deudas	53
Figura 4.5: Modelo de Caso de Uso Realizar Pago Fraccionado de Alumnos.....	54
Figura 4.6: Modelo de Caso de Uso Reporte de Deudas	55
Figura 4.7: Modelo de Caso de Uso Reporte de Cobranzas	56
Figura 4.8: Modelo de Caso de Uso Administración de Usuarios.....	57
Figura 4.9: Modelo de Caso de Uso Ingreso de Usuarios.....	57
Figura 4.10: Modelo de Caso de Uso Modificar datos de Usuarios	58
Figura 4.11: Modelo de Caso de Uso Eliminar Usuarios	59
Figura 4.12: Modelo de Casos de Uso del Sistema	60
Figura 4.13: Modelo de Actores	70
Figura 4.14: Inteface Ingreso de Deudas de Alumnos	71
Figura 4.15: Inteface Realizar Pago de Alumno	71
Figura 4.16: Inteface Realizar Pago Fraccionado de Alumnos.....	71
Figura 4.17: Inteface Consulta de Cobranzas	72

Figura 4.18: Inteface Consulta de Alumnos con Deudas.....	72
Figura 4.19: Clases que manejan la Lógica del Negocio.....	72
Figura 4.20: Diagrama de Clases	73
Figura 4.21: Paquete de Dominio	73
Figura 4.22: Diagrama de Despliegue.....	74
Figura 4.23: Capas del Sistema.....	75
Figura 4.24: Diagrama de Componentes.....	76
Figura 4.25: Diagrama de Colaboración - Realizar Pago de Alumnos	76
Figura 4.26: Diagrama de Colaboración - Realizar Pago Fraccionado de Alumnos	77
Figura 4.27: Diagrama de Estados - Realizar Pago de Alumno.....	77
Figura 4.28: Diagrama de Secuencias - Realizar Pago de Alumno	78
Figura 4.29: Diagrama de Estados - Realizar Pago Fraccionado de Alumnos	79
Figura 4.30: Diagrama de Secuencias - Realizar Pago Fraccionado de Alumnos.....	80
Figura 4.31: Diseño Físico - Diagrama Entidad Relación	81
Figura 4.32: Configuración del ODBC	82
Figura 4.33: Configuración base datos	82
Figura 4.34: Ingreso de Login.....	83
Figura 4.35: Menú Principal.....	83
Figura 4.36: Ingresar Deudas de Alumnos.....	84
Figura 4.37: Cobranza del Pago del Alumno.....	85
Figura 4.38: Modificación de la Deuda del Alumno	86
Figura 4.39: Cobranza del Pago Fraccionado	87
Figura 4.40: Reporte de Cobranzas.....	88
Figura 4.41: Reporte de Alumnos con Deudas	88
Figura 4.42: Listado General de Cobranzas.....	89

Figura 4.43: Reporte de Pagos de Morosos	89
Figura 4.44: Listado de Deudas	90
Figura 4.45: Reporte de Deudas Acumuladas.....	90
Figura 4.46: Administración de Usuarios	91
Figura 4.47: Administración de Alumnos.....	92
Figura 4.48: Administración de Departamento.....	93

ÍNDICES DE CUADROS

CUADRO	PÁG
Cuadro 2.6: Esquema de la Metodología del Sistema	20
Cuadro 3.1: Comparativa entre las Metodologías.....	23
Cuadro 3.2: Ponderación de las Herramientas de Desarrollo	31
Cuadro 3.3: Comparación entre Base de Datos	36
Cuadro 3.3.5: Esquema del Desarrollo del Sistema.....	38
Cuadro 4.1: Esquema Input del Sistema	39
Cuadro 4.5.1: Servidor Web	74
Cuadro 4.5.2: Servidor de Base de Datos	74
Cuadro 4.5.3: Estaciones Cliente	74

INTRODUCCIÓN

La idea central de este trabajo se orienta a analizar, diseñar, construir e implementar el sistema de Pago y Gestión de Cobranza Vía Web de la Facultad de Ingeniería de Sistemas de UNMSM, se plantea como estrategia de implementación la gradualidad, lo cual significa atender primero a los estudiantes de Pre Grado que mantengan alguna deuda a la FISI y luego a otros usuarios como los egresados, bachilleres, etc. que deseen realizar algún trámite que genere un pago de una deuda a la FISI, por ejemplo: postular a la Maestría, etc.

Es necesario que se brinde a los usuarios, información de los estados de las deudas y cobranzas a través del uso de Internet, de manera que se facilite esta labor, lo cual traerá consigo beneficios tanto para los usuarios quienes se beneficiarán con un mejor servicio y la Facultad podrá verificar dinero de sus usuarios morosos.

ANTECEDENTES

Las universidades sobre todo las particulares han incrementado sus ingresos en esta última década gracias a la mejora de sus procesos administrativos los cuales han sido optimizados y automatizados con el uso de las Tecnologías de Información y Comunicación, de manera que puedan efectuar el control de los pagos y deudas de sus usuarios.

Existe en nuestra Facultad la necesidad de contar con un sistema web que permita efectuar un buen seguimiento y control sobre las deudas, con el propósito de incrementar la eficiencia en los pagos, para lograr una óptima función de cobranza, permitiendo con ello que la FISI obtenga mayor liquidez.

No es raro ver que varias de las universidades en el Perú se valen de estos medios para agilizar sus procesos, para atender a dichos usuarios (alumnos, padres y/o tutores, etc.) en la ejecución de diversas operaciones administrativas y financieras.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 EL PROBLEMA

Actualmente nos toma mucho tiempo buscar en el Sistema Académico-Administrativo la información que se requiere para efectuar la consulta y posterior pago de deudas pendientes; esto debido a que no se cuenta con un sistema de información y la base de datos que permitan darle el tratamiento requerido.

1.1.1 PROBLEMA PRINCIPAL

La carencia de un Sistema automatizado de información y la correspondiente Base de Datos, impide realizar una óptima labor de control de deudas impagas y posterior cobranza de las mismas.

1.1.2 PROBLEMA SECUNDARIO

- Determinar los medios para que se implemente el Sistema de Pago y Gestión de Cobranza para la FISI de la Universidad Nacional Mayor de San Marcos.

1.2 DELIMITACIÓN ESPACIAL

El proyecto se encuentra dentro del contexto UNMSM-Facultad de Ingeniería de Sistemas, definiendo o seleccionado la mejor plataforma tecnológica. Por lo tanto se realizará prototipos web cumpliendo todos los objetivos antes expuestos.

1.3 JUSTIFICACIÓN

El protagonismo del nuevo escenario en la educación, con las nuevas tecnologías web en la comunicación de información on-line para conseguir resultados en tiempos muy cortos y cubrir distancias. Estos deben estar basados en los resultados de la oportunidad y calidad de la información, esto tiene un efecto para que el alumno realice a tiempo sus pagos, las cuales expresan su puntualidad para cumplir sus obligaciones económicas. Esta actitud favorable hacia el cumplimiento, le permitirá al estudiante colaborar con su Facultad en la recuperación de ingresos propios.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

El objetivo es desarrollar una propuesta para la construcción de un Sistema de Pago y Gestión de Cobranza Vía Web en la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, con un modelo propio y haciendo uso de las tecnología Web para el acceso desde internet.

1.4.2 OBJETIVOS ESPECÍFICOS

- Registrar, controlar y emitir los resultados de los pagos y cobranzas realizadas en un portal web, actualizando a nivel web la información más reciente.
- Recortar los plazos de pagos de los alumnos y sus procesos administrativos vinculados a matrícula.

1.5 ORGANIZACION DE LA TESINA

La tesina se divide en 4 capítulos, se ha organizado de la siguiente manera:

En el segundo capítulo se tratará todo el marco teórico sobre el cual se sustenta el desarrollo de esta tesis la identificación de las metodologías rup, scrum, xp y el marco individual y legal de la tesina.

En el tercer capítulo se tratará todo el estado del arte, benchmarking de las metodologías rup, scrum y xp, selección de la herramienta de desarrollo, benchmarking entre las herramienta de desarrollo, comparación entre base de datos.

En el cuarto capítulo se tratará todo el análisis y diseño del sistema, se mostrará los casos de uso, el diseño de la base de datos y prototipo del sistema.

CAPÍTULO II: MARCO TEÓRICO

En este capítulo describiremos las principales características de las metodologías Rup, Scrum y Xp detallando sus conceptos básicos de cada metodología.

2.1 METODOLOGÍA RUP

Según [JACOBSON,+,2000] afirma lo siguiente: “RUP es un marco de trabajo genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones y diferentes tamaños de proyectos”.

2.1.1 PROCESO DIRIGIDO POR CASOS DE USO

Según [KRUCHTEN,2000] “Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.”

Según [KRUCHTEN,2000] “En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba”. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la Figura 2.1.

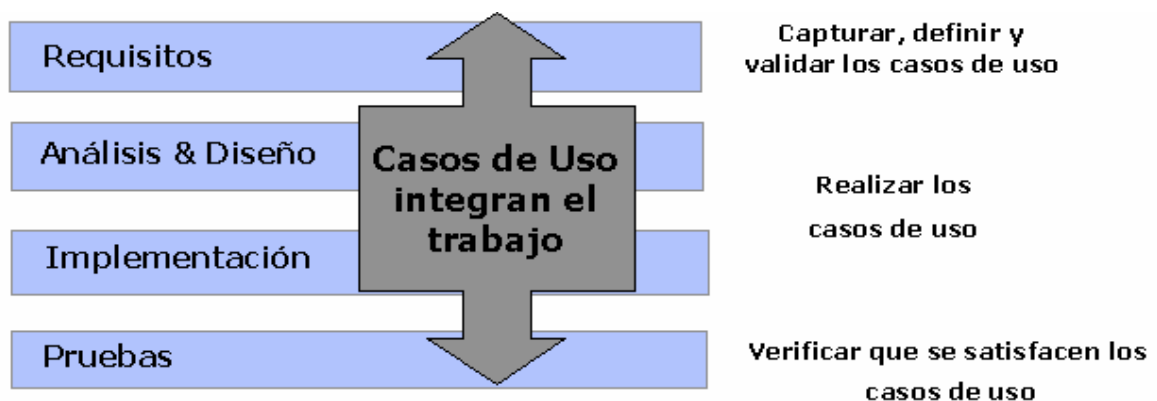


Figura 2.1: Los Casos de Uso integran el trabajo Según [KRUCHTEN,2000]

2.1.2 PROCESO CENTRADO EN LA ARQUITECTURA

Según [KRUCHTEN,2000] “La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.”

Según [KRUCHTEN,2000] “Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de *baselines* y se va modificando dependiendo de las necesidades del proyecto.”

2.1.3 PROCESO ITERATIVO E INCREMENTAL

Según [JACOBSON,+,2000] “El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.”

2.1.4 ESTRUCTURA DE RUP

Según [KRUCHTEN,2000] “El proceso puede ser descrito en dos dimensiones o ejes”:

Eje horizontal: Según [KRUCHTEN,2000] “Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la

Figura 2.2 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.”

Eje vertical: Según [KRUCHTEN,2000] “Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles”

En la Figura 2.2 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

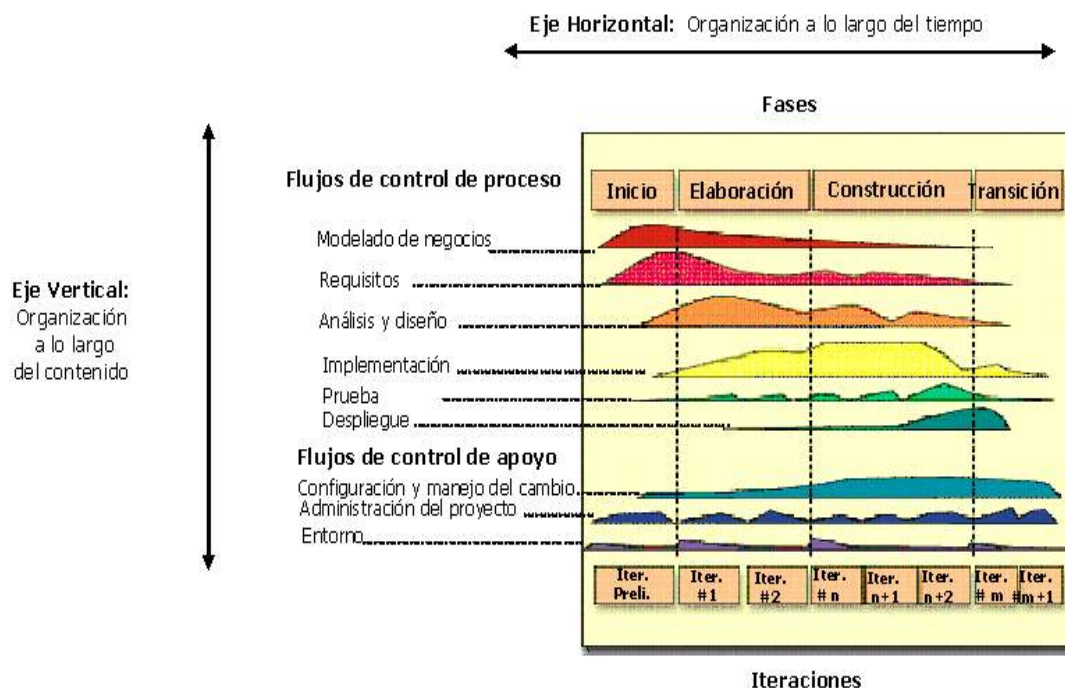


Figura 2.2: Estructura de RUP Según [KRUCHTEN,2000]

2.1.5 ESTRUCTURA DINÁMICA DEL PROCESO

Según [KRUCHTEN,2000] “RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.”

2.1.5.1 Inicio

Según [KRUCHTEN,2000] Los objetivos de esta fase son:

- “Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.
- Estimar los riesgos, las fuentes de incertidumbre”.

2.1.5.2 Elaboración

Según [KRUCHTEN,2000] Los objetivos de esta fase son:

- “Definir, validar y cimentar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.”

2.1.5.3 Construcción

Según [KRUCHTEN,2000] Los objetivos concretos según incluyen:

- “Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.”

2.1.5.4 Transición

Según [KRUCHTEN,2000] se citan algunas de las cosas que puede incluir esta fase:

- “Prueba de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto.
- Conversión de las bases de datos operacionales.
- Entrenamiento de los usuarios y técnicos de mantenimiento.
- Traspaso del producto a los equipos de marketing, distribución y venta.”

2.2 METODOLOGÍA SCRUM

Según [PALACIO,2007] “Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque la gestión no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.”

Según [PALACIO,2007] Scrum es una metodología ágil:

- “Es un modo de desarrollo de carácter adaptable.
- Orientado a las personas antes que a los procesos.
- Emplea desarrollo ágil: iterativo e incremental.”

2.2.1 DESARROLLO INCREMENTAL

Según [PALACIO,2007] “En el proyecto, no se trabaja con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.”

2.2.1.1 Desarrollo evolutivo

Según [PALACIO,2007] “Con Scrum, el diseño y la estructura del resultado se construyen de forma evolutiva. No se considera que la descripción detallada del

producto, del servicio, de la estrategia o de la arquitectura del software (según el caso) deban realizarse en la primera *fase* del proyecto.”

2.2.1.2 Auto-organización

Según [PALACIO,2007] “La gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos. En Scrum los equipos son auto-organizados, con margen de decisión suficiente para tomar las decisiones que consideren oportunas.”

<

Según [PALACIO,2007] “Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo, que es necesaria y debe basarse en la colaboración abierta entre todos según los conocimientos y capacidades de cada persona, y no según su rol o puesto.”

2.2.2 VISIÓN GENERAL DEL PROCESO

Según [PALACIO,2007] “Al comenzar cada iteración (sprint) se determina qué partes se van a construir, tomando como criterios la prioridad para el negocio, y la cantidad de trabajo que se podrá abordar durante la iteración.”

En la Figura 2.3 se muestra a la iteración (sprint) del Scrum como base del desarrollo iterativo e incremental.

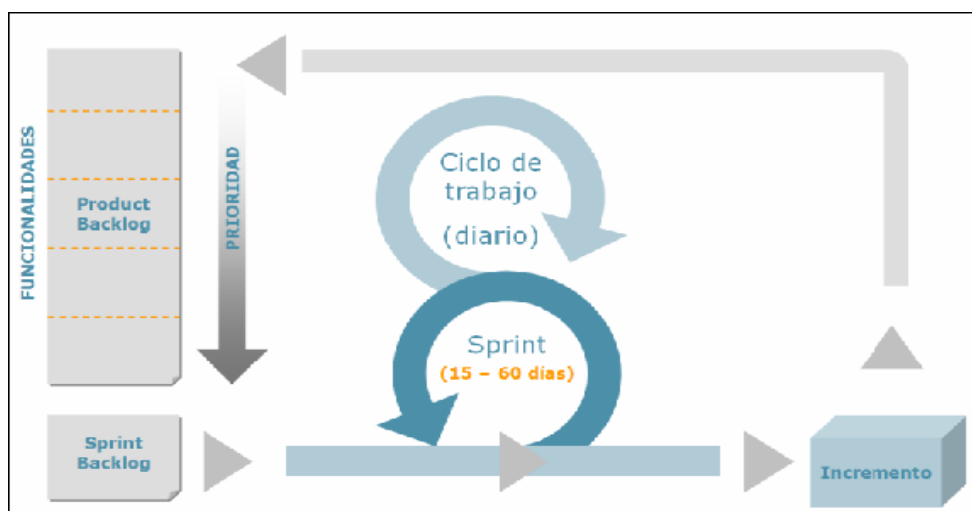


Figura 2.3: Iteraciones del Scrum Según [PALACIO,2007]

Según [PALACIO,2007] Los componentes y conceptos empleados en Scrum son:

2.2.2.1 Las Reuniones

Planificación del sprint: Según [PALACIO,2007] “Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál es el trabajo y los objetivos que se deben cubrir con esa iteración.”

Seguimiento del sprint: Según [PALACIO,2007] “Breve reunión diaria para dar repaso al avance de cada tarea, y al trabajo previsto para la jornada.

Revisión de sprint: Según [PALACIO,2007] “Análisis y revisión del incremento generado. Esta reunión no debe tomarse como un “acontecimiento especial”, sino como la presentación normal de los resultados.”

En la Figura 2.4 se muestra como se divide las tres reuniones del Scrum.

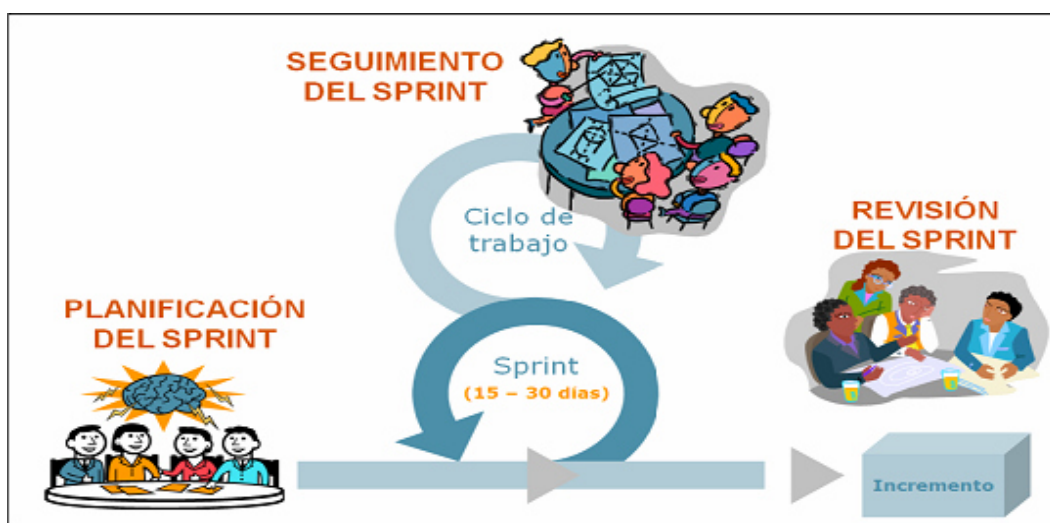


Figura 2.4: Reuniones en Scrum Según [PALACIO,2007]

2.2.2.2 Los Elementos

Product backlog: Según [PALACIO,2007] “Requisitos del sistema. Se parte de la visión del resultado que se desea obtener; y evoluciona durante el desarrollo.”

Sprint Backlog: Según [PALACIO,2007] “Lista de los trabajos que realizará el equipo durante el sprint para generar el incremento previsto.”

Incremento: Según [PALACIO,2007] “Resultado de cada sprint. Se trata de un resultado completamente terminado y en condiciones de ser usado.”

2.2.2.3 Los Roles o Responsabilidades

Según [PALACIO,2007] El grado de funcionamiento de Scrum en la organización depende directamente de estas tres condiciones del libro [PALACIO,2007]:

- “Características del entorno (organización y proyecto) adecuadas para desarrollo ágil.
- Conocimiento de la metodología de trabajo en todas las personas de la organización y las implicadas del cliente.
- Asignación de responsabilidades:

(Del producto, Del desarrollo, Del funcionamiento de Scrum)”

2.2.3 SCRUM MANAGEMENT

Según [PALACIO,2007] “La adopción de una scrum management será exitosa en la medida del nivel de competencia y responsabilidad que ofrezca la organización a su implantación en tres áreas: management o gestión de la organización, calidad o procesos, y producción”

En la Figura 2.5 se muestra las tres áreas del Scrum Management.



Figura 2.5: Scrum Management Según [PALACIO,2007]

2.3 LA PROGRAMACIÓN EXTREMA

Según [ANAYA,2007] “Metodología ágil basada en cuatro principios: simplicidad, comunicación, retroalimentación y valor. Además, orientada por pruebas y refactorización, se diseña e implementan las pruebas antes de programar la funcionalidad, el programador crea sus propios tests de unidad. El objetivo de Xp son grupos pequeños y medianos de construcción de software en donde los requisitos aún son muy ambiguos, cambian rápidamente o son de alto riesgo. Xp busca la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto. Además, sugiere que el lugar de trabajo sea una sala amplia, si es posible sin divisiones (en el centro los programadores, en la periferia los equipos individuales). Una ventaja del espacio abierto es el incremento en la comunicación y el proporcionar una agenda dinámica en el entorno de cada proyecto.”

2.3.1 ACTIVIDADES DE XP

2.3.1.1 Codificar

Según [ANAYA,2007] “Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.”

2.3.1.2 Hacer pruebas

Según [ANAYA,2007] “Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.”

2.3.1.3 Escuchar

Según [ANAYA,2007] “Los programadores no lo conocemos todo, y sobre todo muchas cosas que las personas de negocios piensan que son interesantes. Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quien necesita la información.”

2.3.1.4 Diseñar

Según [ANAYA,2007] “El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.”

2.3.1.5 Prácticas Básicas de XP

Según [ANAYA,2007] “De forma aislada, cualquier práctica individual de Xp tiene poco sentido, pero en conjunto, unas compensan las carencias que las otras puedan tener. Para evaluar Xp se tiene que mirar el conjunto de prácticas”:

- El juego de la Planificación (Planning Game): Según [ANAYA,2007] “El alcance de la siguiente versión esta definido por las consideraciones de negocios (prioridad de los módulos, fechas de entrega) y estimaciones técnicas (estimaciones de funciones, consecuencias).”
- Versiones Pequeñas (Short Releases): Según [ANAYA,2007] “Un sistema simple se pone rápidamente en producción. Periódicamente, se producen nuevas versiones agregando en cada iteración aquellas funciones consideradas valiosas para el cliente.”

- Metáfora del Sistema (Metaphor): Según [ANAYA,2007] “Cada Proyecto es guiado por una historia simple de cómo funciona el sistema en general, reemplaza a la arquitectura y debe estar en lenguaje común, entendible para todos (Cliente y Desarrolladores), esta puede cambiar permanentemente.”
- Diseño Simple (Simple Designs): Según [ANAYA,2007] “El sistema se diseña con la máxima simplicidad posible. Se plasma el diseño en tarjetas CRC (Clase – Responsabilidad - Colaboración), no se implementan características que no son necesarias.”
- Pruebas Continuas (Testing): Según [ANAYA,2007] “Los casos de prueba se escriben antes que el código. Los desarrolladores escriben pruebas unitarias y los clientes especifican pruebas funcionales.”
- Refactorización (Refactoring): Según [ANAYA,2007] “Es posible reestructurar el sistema sin cambiar su comportamiento, si el código se está volviendo complicado se debería modificar el diseño y volver a uno más simple.”
- Programación por parejas (Pair Programming): Según [ANAYA,2007] “El código es escrito por dos personas trabajando en el mismo computador.”
- Posesión Colectiva del Código (Collective Code Ownership): Según [ANAYA,2007] “Cualquier programador puede cambiar cualquier parte del sistema en cualquier momento, siempre se utilizan estándares y se excluyen los comentarios.”
- Cliente en el Sitio (On Site Customer): Según [ANAYA,2007] “El equipo de desarrollo tiene acceso todo el tiempo al cliente, el cual está disponible para responder preguntas, fijar prioridades, etc.”

- Integración continua (Continuous Integration): Según [ANAYA,2007]
“Los cambios se integran en el código base varias veces por día. Todos los casos de prueba se deben pasar antes y después de la integración, se dispone de una máquina para la integración y se realizan test funcionales en donde participa el cliente.”
- Semana laboral de 40 horas (40-Hour Week): Según [ANAYA,2007]
“Cada Trabajador trabaja no más de 40 Horas por semana. Si fuera necesario hacer horas extra, esto no debería hacerse dos semanas consecutivas.”
- Estándares de Codificación (Coding Standard): Según [ANAYA,2007]
“Todo el código debe estar escrito de acuerdo a un estándar de codificación”

2.3.2 CICLO DE VIDA XP

Según [ANAYA,2007] “El ciclo de vida de Xp se enfatiza en el carácter interactivo e incremental del desarrollo, una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas que en el caso de Xp corresponden a un conjunto de historias de usuarios.”

Según [ANAYA,2007] “Las iteraciones son relativamente cortas ya que se piensa que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial orientada a programar las iteraciones de desarrollo y cada iteración incluye diseño, codificación y pruebas, fases superpuestas de tal manera que no se separen en el tiempo.”

La siguiente figura 2.6 Según [ANAYA,2007] muestra las fases en las que se subdivide el ciclo de vida Xp:

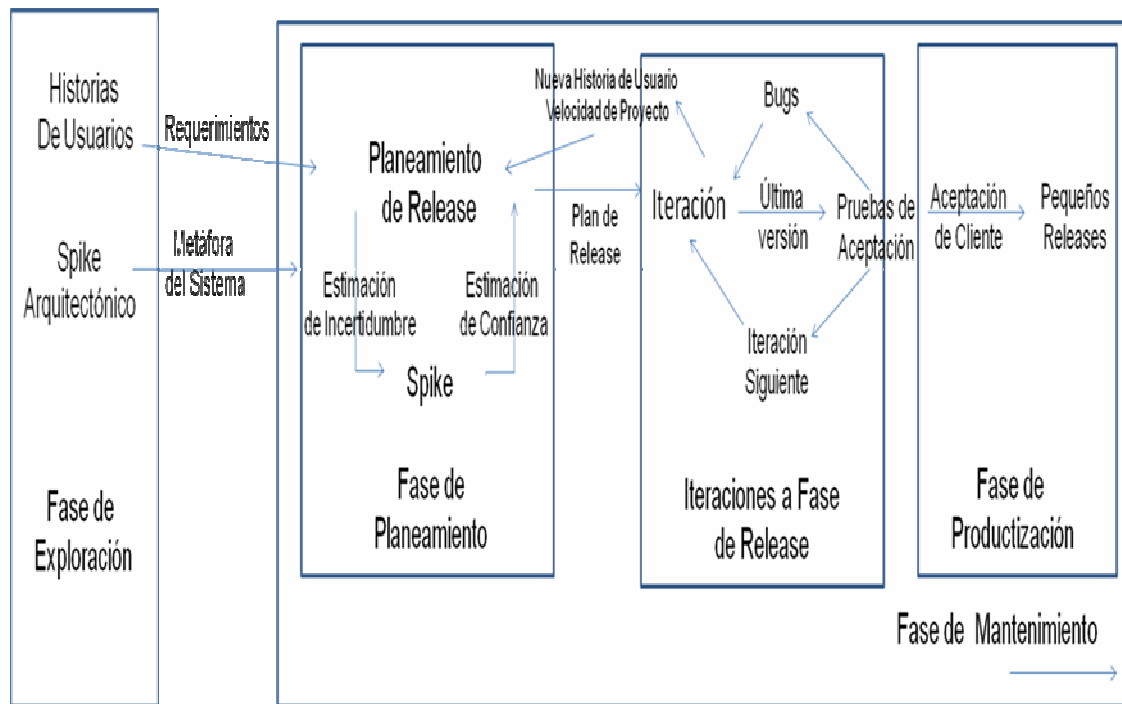


Figura 2.6: Ciclo de vida de la Programación Extrema Según [ANAYA,2007]

2.3.2.1 Fase de la exploración: Según [ANAYA,2007] “En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto”.

2.3.2.2 Fase del planeamiento: Según [ANAYA,2007] “Se priorizan las historias de usuario y se acuerda el alcance del release. Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. La fase de planeamiento toma un par de días. El cronograma fijado en la etapa de planeamiento se realiza a un número de iteraciones, cada una toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo.”

2.3.2.3 Fase de producción: Según [ANAYA,2007] “Requiere prueba y comprobación extra del funcionamiento del sistema antes de que éste se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en el release actual. Durante esta fase, las iteraciones pueden ser aceleradas de una a tres semanas. Las ideas y las sugerencias pospuestas se documentan para una puesta en práctica posterior por ejemplo en la fase de mantenimiento.”

2.3.2.4 Fase de mantenimiento: Según [ANAYA,2007] “Requiere de un mayor esfuerzo para satisfacer también las tareas del cliente. Así, la velocidad del desarrollo puede desacelerar después de que el sistema esté en la producción.”

2.3.2.5 Fase de muerte: Según [ANAYA,2007] “Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.”

2.4 MARCO INDIVIDUAL

Se dispone presentar prototipos vía web del sistema de pagos de manera que se puedan mostrar los procesos internos de pagos de los alumnos, como las cobranzas del área de economía. Una información clara y oportuna en los requerimientos de los pagos de los alumnos, está es de importancia para los alumnos para sus trámites académicos y necesario para la reducción de la brecha de morosidad.

2.5 MARCO LEGAL

La adquisición y las licencias del software base, así como los permisos correspondientes para evitar posibles sanciones del órgano regulador.

2.6 ESQUEMA DE LA METODOLOGÍA DEL SISTEMA

Se presenta un esquema de la metodología del sistema con la cual se trabajará en Fases RUP y sus actividades, dentro del modulo del sistema.

	METODOLOGIA RUP	
	Intercepción (puesta en marcha)	Elaboración (definición, análisis, diseño)
Fases RUP Actividades RUP		
Modelado del negocio	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Análisis de requisitos	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Gestión de configuración y cambios	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Gestión del proyecto	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes

Cuadro 2.6 Esquema de la Metodología del Sistema Elaboración Propia

CAPÍTULO III: ESTADO DEL ARTE

3.1 BENCHMARKING ENTRE METODOLOGÍAS EXISTENTES

Según [LAGOS, 2008] “Las comparaciones entre metodologías encontradas en las diversas bibliografías, no tiene como fin asegurar si una metodología es mejor que la otra, tal es el caso en la presente investigación, en donde la finalidad de una comparación es establecer las diferencias entre cada metodología en función de ciertas características que se plantean, y poder concluir en base a estas diferencias las ventajas y/o desventajas de cada metodología de acuerdo al tipo de sistema a realizar.”

3.1.1 ANÁLISIS DE LOS CRITERIOS DE EVALUACIÓN

Según [LAGOS, 2008] “Para realizar el análisis entre las metodologías para el desarrollo de software se propone el uso de 3 parámetros”:

3.1.1.1 Sistema como algo cambiante: Este parámetro implica que el equipo de desarrollo tome como ventaja el cambio de desarrollo, pues además que aprende más, genera una mayor satisfacción en el cliente. Este parámetro también aplica para la estructura del software, pues no debe ser muy costoso el implementar los cambios.

3.1.1.2 Colaboración entre los miembros del equipo: Entre los miembros del equipo debe existir una constante comunicación, cara a cara de preferencia pues es el método más efectivo, pues si bien puede existir comunicación, esta no contiene todo.

3.1.1.3 Características Metodológicas (CM): Identificación de Características Metodológicas en la evaluación de metodologías.

- **Resultados:** Según [LAGOS, 2008] “Verifica el grado de coincidencia entre estimaciones realizadas y el tiempo real dedicado al proyectos, se pueden medir los resultados de cada iteración dependiendo de la metodología de desarrollo”. Esto para proporcionar un resultado completo sobre producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental.

- **Simplicidad:** Según [LAGOS, 2008] “Implica optar por los caminos más simples, para que cumplan fielmente los requisitos y que brinden calidad, esto facilitará la adaptación a los cambios”.

La simplicidad se basa en el entendimiento profundo del asunto que se quiere transmitir y en la capacidad de hacerlo de una forma clara y concisa, es decir en conseguir el máximo efecto de comunicación con la mínima energía. Una Simplicidad organizada son parte de los sistemas vivientes, cuya característica es que son abiertos. La simplicidad tiene algunas ventajas notables como son: Proximidad, Reconocibilidad, Inmediatez, Usabilidad.

- **Adaptabilidad:** Todo proyecto debe en lo posible permitir la adaptación de nuevos requisitos. La adaptabilidad Es la propiedad que tiene un sistema de aprender y modificar un proceso, un estado o una característica de acuerdo a las modificaciones que sufre el contexto. Esto se logra a través de un mecanismo de adaptación que permita responder a los cambios internos y externos a través del tiempo.

- **Excelencia técnica:** Las mejores arquitecturas, requisitos y diseños surgen de equipos organizados de acuerdo a la necesidad. La excelencia técnica es vital tanto para el equipo como para el proyecto, esto es, decisiones adecuadas y oportunidad en la toma de las mismas, habilidad en el manejo de algunas técnicas y buen juicio para proceder.

· **Prácticas de colaboración:** Se propone que exista una interacción constante entre el equipo de desarrollo y los empleados. Asimismo, asegura que el trabajo en equipo repercuta directamente en la productividad y en el compromiso por la organización. El Jefe como empleados incrementen la cultura de colaboración en el marco de su propia organización para el beneficio del proyecto. La colaboración no es una simple tecnología, proceso o servicio, sino un valor que puede ser asociado a diversos procesos e interacciones.

A continuación mostraremos un cuadro más detallado donde se tomara las prioridades del 1 al 5, para el análisis con el uso de los 3 parámetros de evaluación: Sistema como algo cambiante, Colaboración entre los miembros del equipo y Características metodológicas (Resultado, Simplicidad, Adaptabilidad, Excelencia Técnica y Prácticas de Colaboración) indicando si el método cumple con mayor o menor prioridad el parámetro, respectivamente.

	TRADICIONALES	ÁGILES	
	RUP	XP	SCRUM
Sistema con algo cambiante	4	5	5
Colaboración	5	5	5
Características Metodológicas			
- Resultado	5	5	5
- Simplicidad	5	5	5
- Adaptabilidad	3	3	4
- Excelencia Técnica	5	4	3
- Prácticas de Colaboración	5	5	4
MEDIA Características Metodológicas	4,6	4,4	4,2
MEDIA TOTAL	4,57	4,57	4,43

Cuadro 3.1: Comparativa entre las Metodologías, Según [LAGOS, 2008]

El conjunto de metodologías se ubican cerca en puntaje, el XP y RUP, pero sin embargo, se eligió el proceso unificado de Rational (Rational Unified Process), para este proyecto pues satisface a nuestro desarrollo en mayor medida.

3.2 SELECCIÓN DE LA HERRAMIENTA

3.2.1 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE

3.2.1.1 PHP (HyperText PreProcesador)

Según [CARRILLO, 2007] expresa lo siguiente:

“PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*). PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI”.

Según [CARRILLO, 2007] expresa lo siguiente:

“Ventajas

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial <http://www.php.net/manual/es/>, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable”.

3.2.1.2 J2EE (Java 2 Enterprise Edition)

Según autor [CARRILLO, 2007] expresa lo siguiente:

“J2EE son las siglas de Java 2 Enterprise Edition que es la edición empresarial del paquete Java creada y distribuida por Sun Microsystems. Comprenden un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales. Debido a que J2EE no deja de ser un estándar, existen otros productos desarrollados a partir de ella aunque no exclusivamente. Para el desarrollo de estas aplicaciones, J2EE permite implementar todas las capas en qué consiste una solución empresarial como son: La capa del cliente, la capa de presentación, la capa de negocios, la capa de integración y la capa de recursos.

Los beneficios de esta arquitectura vienen de la mano de los beneficios directos que produce Java. Estos beneficios son: la seguridad, el uso de la orientación a objetos, la disponibilidad en múltiples plataformas, etc.

La arquitectura J2EE conlleva a que la petición de un cliente se dirija hacia un servidor Web donde existe un Contenedor de Servlets, aquí se procesa la petición y permite llevar el resultado al usuario según su necesidad. Esta distribución de funcionalidades permite que la arquitectura de J2EE pueda contener amplias seguridades y alternativas de desarrollo. Algunas alternativas de desarrollo al estándar J2EE, las cuales brindan ventajas en el diseño e incluso en la Implementación, se listan a continuación”:

La arquitectura MVC

Según autor [CARRILLO, 2007] expresa lo siguiente: “La arquitectura *Model-View-Controller* surgió como patrón arquitectónico para el desarrollo de interfaces gráficas de usuario en entornos Smalltalk. Su concepto se basaba en separar el modelo de datos de la aplicación de su representación de cara al usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.”

ORM (Object Relational Mapping) Mapero Objeto-Relacional

Según autor [CARRILLO, 2007] expresa lo siguiente: “Es un modelo que permite el manejo de datos desde una arquitectura relacional (Base de Datos), a una arquitectura orientada a objetos. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Hay paquetes comerciales y de uso libre disponibles que desarrollan el mapeo relacional de objetos, aunque algunos programadores prefieren crear sus propias herramientas ORM”.

Inversión de Control (IoC, *Inversion of Control*)

Según autor [CARRILLO, 2007] expresa lo siguiente: “Es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones. En la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. Se considera como alternativa a arquitectura del estándar Enterprise Java Beans (EJB’s) hasta la versión 2.x”.

La Programación Orientada a Aspectos (POA)

Según autor [CARRILLO, 2007] expresa lo siguiente: “Es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Gracias a la POA se pueden encapsular los diferentes conceptos que componen una aplicación en entidades bien definidas, eliminando las dependencias entre cada uno de los módulos. De esta forma se consigue razonar mejor sobre los conceptos, se elimina la dispersión del código y las implementaciones resultan más comprensibles, adaptables y reusables”.

3.2.1.3 ASP (Active Server Pages)

Según autor [CARRILLO, 2007] expresa lo siguiente: “Un archivo de páginas Active Server (ASP) es un archivo de texto con la extensión .asp que contiene cualquier combinación de Texto, Etiquetas HTML y Secuencias de comandos del servidor.

ASP es un lenguaje de programación sobre páginas Web creado por Microsoft con el fin de brindar un rápido aprendizaje y fácil creación de sitios Web dinámicos. El contenedor de este tipo de páginas se encuentra embebido en el IIS (Internet Information Service) el cual es el motor Microsoft para la publicación Web.

Este lenguaje permite que el código sea incrustado en un documento HTML y que corra en el servidor. Al igual que PHP, es necesario el uso de scripts y el código no será indispensable hasta que se reciba una solicitud del cliente. El código fuente nunca deja el servidor haciendo por lo tanto a ASP muy seguro y fácil de programar. ASP permite también el manejo de Scripts sobre el cliente, tal es el caso de VBScript, y de Jscript. JavaScript no es soportado”.

3.2.1.4 .NET

Según autor [CARRILLO, 2007] expresa lo siguiente: “.NET es una plataforma de desarrollo de software desarrollada por Microsoft con énfasis en la transparencia de redes que permita un rápido desarrollo de aplicaciones. Basado en esta tecnología, Microsoft intenta realizar un desarrollo horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de desarrollo. Se pueden considerar a .NET como una respuesta de Microsoft al creciente mercado de los negocios entornos Web, como una competencia a la plataforma Java de Sun Microsystems.

.NET plantea el uso de Servicios Web para reusabilidad de servicios a diferencia de los Servlets de J2EE. Así mismo, .NET plantea el uso de archivos XML, para la compartición de Datos. .NET intenta ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones o como la misma plataforma las denomina, soluciones, permitiendo a su vez una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo. El problema es el coste que requiere el manejo de herramientas Microsoft, así como el soporte y la utilización de una sola tecnología computacional envuelta en el sistema operativo Microsoft Windows.”

3.3 BENCHMARKING ENTRE LAS HERRAMIENTAS DE DESARROLLO

Las comparaciones entre las herramientas de desarrollo no tiene como fin asegurar si una herramienta de desarrollo es mejor que la otra, tal es el caso en la presente investigación, en donde la finalidad de una comparación es establecer las diferencias entre cada herramienta de desarrollo en función de ciertas características que se plantean, y poder concluir en base a estas diferencias las ventajas y/o desventajas de cada herramienta de desarrollo de acuerdo al tipo de sistema a realizar.

3.3.1 ANÁLISIS DE LOS CRITERIOS DE EVALUACIÓN

Para realizar el análisis entre las diversas plataformas de desarrollo proponemos el uso de 5 parámetros, pues se eligieron para este proyecto como los criterios más apropiados.

Independencia de la Plataforma: Según [QUINN,2002] “Independencia de la plataforma, dentro de nuestras posibilidades, las páginas Web deberán ser independientes de cualquier plataforma, lo que significa que ellas serán accesibles sin tomar en cuenta la plataforma del usuario ni su configuración”.

Escalabilidad: Según [PICON,2007] “Es aumentar la capacidad de atender usuarios o volumen de datos de manera lineal con la capacidad de cómputo añadida. La Escalabilidad vertical es hacia arriba, “upgrades” a cada nodo y la Escalabilidad horizontal (hacia fuera) es aumentar el número de nodos”.

Robustez de la tecnología: Solidez de la tecnología utilizada (tecnología probada y/o disponible a nivel comercial). La robustez del lenguaje de programación se refleja cuando se desempeña adecuadamente bajo cualquier circunstancia.

Velocidad: La velocidad de los sitios web en muchos casos influye sustancialmente en las “páginas vistas” (page views) de un proyecto web en específico y cuál es el rendimiento real de su servidor. La carga de uno o varios sitios web, la velocidad se mide en cuanto debe tardarse en mostrarse las páginas web.

Interfaz Gráfica de Usuario: Es un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. El proceso general para analizar y diseñar una interfaz de usuario empieza con la creación de diferentes modelos de función del sistema.

Simplicidad del Lenguaje: Un lenguaje de programación debe esforzarse en la simplicidad sintáctica y semántica. La simplicidad en la semántica implica que el lenguaje contiene un mínimo número de conceptos y estructuras.

A continuación mostraremos un cuadro más detallado donde se tomara las prioridades del 1 al 5, para el análisis con el uso de los 6 parámetros de evaluación: Independencia de la Plataforma, Escalabilidad, Robustez de la Tecnología, Velocidad, Interfaz Gráfica de Usuario, Simplicidad del Lenguaje indicando si el método cumple con mayor o menor prioridad el parámetro, respectivamente. Definiéndose como la puntuación más alta a 5 y como la más baja a 1.

La herramienta seleccionada para el desarrollo del proyecto tuvo fundamento en la Tabla Comparativa de Herramientas de Desarrollo. Los principales criterios de selección de la herramienta fueron:

Herramienta	J2EE	PHP	.NET	ASP
Independencia de la Plataforma	4	3	2	2
Escalabilidad	4	3	4	3
Robustez de la Tecnología	5	5	4	4
Velocidad	4	5	4	5
Interfaz Gráfica de Usuario	3	3	4	5
Simplicidad del Lenguaje	3	3	4	5
Total	23	22	22	24

**Cuadro 3.2: Ponderación de las Herramientas de Desarrollo,
Según [CARRILLO,2007]**

Los valores anteriormente expuestos muestran el rendimiento de las herramientas basado en experiencias de desarrollo de software personales, por las características definidas la selección fue por el Lenguaje de desarrollo ASP pues permite un desarrollo más rápido y sustentable, así como el diseño de las interfaces web que brinda esta tecnología.

3.3.2 SELECCION DE LA BASE DE DATOS

Los sistemas actuales requieren de un sistema de administración de base de datos potente, que permita manipular los datos mediante consultas de manera confiable con un alto nivel de seguridad y manejo en la integridad de datos y capacidad para atender varios procesos que servirán para implementar este trabajo.

3.3.2.1 Microsost SQL server 2005 Express

Según [CARRILLO, 2007] expresa lo siguiente:

“SQL Server 2000 MSDE es el predecesor de Microsoft SQL 2005 Express Edition (SQL server Express), y se basa en SQL Server 2000.

SQL Server 2005 Express Edition es una versión ligera de SQL Server 2005, casi con las mismas características, salvo con algunas limitaciones como no poseer un regulador de trabajo simultaneo, la base de datos tiene una capacidad máxima de 4 GB y la compatibilidad de RAM es 1.”

Según [CARRILLO, 2007] expresa lo siguiente:

“Características

- Optimizador avanzado de consultas
- Maneja tres niveles de seguridad de acceso al código: seguro, acceso externo y no seguro.
- Aprovechamiento de los procedimientos almacenados como capa de abstracción.
- Compatibilidad con esquemas XML
- Suscriptor de réplica transaccional
- Suscriptor de réplica de combinación
- Cliente para SQL server service bróker
- Compatibilidad con transacciones distribuidas”

3.3.2.2 PostGreSQL

Según [CARRILLO, 2007] expresa lo siguiente: “PostGreSQL es un sistema de gestión de base de datos objeto-relacional (ORDBMS), PostGreSQL es una derivación libre (Open Source), y utiliza el lenguaje SQL92/SQL99 PostGreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostGreSQL no es un sistema de gestión de base de datos puramente orientado a objetos.

PostGreSQL está considerado como la base de datos código abierto más avanzada del mundo. PostGreSQL proporciona un gran número de características que normalmente sólo se encontraban en la base de datos comerciales tales como DB2 u Oracle”.

Según [CARRILLO, 2007] expresa lo siguiente:

“Características:

- Implementación del estándar SQL92/SQL99.
- Soporte distintos tipos de datos, y creación de tipos propios.
- Incorpora funciones como: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos en la base de datos.
- MVCC, o Control de Concurrencia Multi-Versión (Multi-Versión Concurrency Control), es tecnología que PostgreSQL usa para evitar bloqueos innecesarios.
- Write ahead logging, incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos, garantizando ante una caída de la base de datos, y presenta un registro de las transacciones que permitirá restaurar la base de datos.”

3.3.2.3 MySQL

Este gestor de base de datos es uno de los más ligeros y potentes que existen para aplicativos Web, en especial con sitios programadores con PHP.

Según [CARRILLO, 2007] expresa lo siguiente:

“Características:

Las características más importantes del software de base de datos MySQL son:

- Corre sobre diferentes plataformas.
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C,C++,Eiffel,Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-threaded mediante threads del kernel.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Permite añadir otro sistema de almacenamiento fácilmente.
- Posee un sistema de reserva de memoria muy rápido basado en threads.
- Las funciones SQL están implementadas usando una librería altamente optimizada.
- Soporte completo para operadores y funciones.
- Sistema de privilegios con contraseñas flexibles y seguras, permite verificación basada en el host. Maneja encriptación de contraseñas.
- Soporte para grandes bases de datos.
- La conexión de clientes con el servidor MySQL usan sockets TCP/IP en cualquier plataforma. En sistemas Unix, los clientes usan ficheros socket.
- Los servidores Windows soportan conexiones con memoria compartida para la version 5.0

- La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC.
- Soporte completo para distintos conjuntos de caracteres.
- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.”

3.3.3 COMPARACION ENTRE BASE DE DATOS

La Tabla muestra un análisis comparativo entre los motores de bases de datos propuestos, según varias características requeridas.

CARACTERÍSTICA	MS SQL Express	PostGreSQL	MySQL
Gratuito	X	X	X
Open Source		X	X
Procedimientos almacenados	X	X	X
Control de concurrencia	X	X	X
Recuperación de fallos	X	X	X

Cuadro 3.3: Comparación entre Base de Datos, Según [CARRILLO,2007]

El conjunto de características definidas se ubican cerca en condiciones, sin embargo para el proyecto utilizaremos MySQL, por ser robusta y ligera, cuya licencia de para su utilización es gratuita, también permite el uso de funciones definidas por el usuario en sentencias Select, Insert o Update que faciliten la obtención de resultados en menos pasos permitiendo en el desarrollo del sistema. Por las características descritas se escoge a MySQL como motor base de datos para el sistema.

3.3.4 RESUMEN DE LAS COMPARACIONES

3.3.4.1 Resumen de la Metodología Elegida

El caso de la presente investigación, podemos concluir en base a las diferencias de las características como las ventajas y/o desventajas de las metodologías, sin embargo se eligió el proceso unificado de Rational (Rational Unified Process), para este proyecto pues satisface a nuestro desarrollo en mayor medida.

3.3.4.2 Resumen de la Herramienta de Desarrollo Elegida

El caso en la presente investigación, podemos concluir en base a estas diferencias las ventajas y/o desventajas de cada herramienta de desarrollo de acuerdo al tipo de sistema a realizar, la cual se mostraron principales criterios y sus valores expuestos para evaluar el resultado, el rendimiento de las herramientas esta basado en experiencias de desarrollo de software personales, por las características definidas la selección fue por el Lenguaje de desarrollo ASP .NET pues permitió un desarrollo más rápido y sustentable.

3.3.4.3 Resumen de la Base de Datos Elegida

El caso en la presente investigación, las características definidas se ubican cerca en condiciones, sin embargo para el proyecto utilizaremos MySQL, por ser robusta y ligera, cuya licencia de para su utilización es gratuita, por esto se escoge a MySQL como motor base de datos para el sistema.

3.3.5 ESQUEMA DEL DESARROLLO DEL SISTEMA

Se presenta un esquema del desarrollo del sistema y la metodología RUP con cual se trabajará en Fases RUP y sus actividades, dentro del modulo del sistema.

Fases RUP Actividades RUP	LENGUAJE: ASP .NET BD: Mysql 5.5
	Construcción (implementación)
Modelado del negocio	
Análisis de requisitos	
Implementación	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Test	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Distribución	
Gestión de configuración y cambios	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Gestión del proyecto	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Gestión del entorno	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes

Cuadro 3.3.5 Esquema del Desarrollo del Sistema Elaboración Propia

CAPÍTULO IV: ANÁLISIS Y DISEÑO DEL SISTEMA

4.1 ESQUEMA INPUT DEL SISTEMA

Fases RUP Actividades	METODOLOGIA RUP		Lenguaje: ASP.NET BD: Mysql 5.5	Transición (fin del proyecto)
	Intercepción (puesta en marcha)	Elaboración (definición, análisis, diseño)		
Modelado del negocio	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	
Análisis de requisitos				
Implementación			Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	
Test			Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Distribución				Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes
Gestión de configuración y cambios	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	
Gestión del proyecto	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes		
Gestión del entorno			Ingreso de Deudas de Alumnos Realizar Pago de Alumno Pago Fraccionado de Alumnos Reportes	

Cuadro 4.1 Esquema Input del Sistema

4.2 VISIÓN

4.2.1 INTRODUCCIÓN

4.2.1.1 Propósito

Llevar las mejores prácticas a la Ingeniería de software, en el modelamiento visual. El presente documento define el marco referencial en el que se desarrolla el proyecto, que consta de la especificación del desarrollo del servicio. Es el propósito del mismo es llevar a cabo a la practica la ingeniería de software, en que consiste el modelamiento visual.

4.2.1.2 Alcance

Este documento detallara el servicio que se va prestar así como el requerimiento del proceso a realizar.

4.2.1.3 Definiciones, Acrónimos, y Abreviaciones

Ver Glosario al final del Documento

4.2.2 POSICIONAMIENTO

Conjunto de servicios básicos y esenciales disponibles en la visión del proyecto que soluciona el problema, brindando dichos servicios de manera fácil, vía internet.

4.2.2.1 Oportunidad de Negocio

Analizando la problemática actual que se vive en la FISI de nuestra universidad, en donde la información económica no esta disponible al momento que es necesaria, hemos encontrado que es posible implementar un conjunto de servicios básicos y esenciales disponibles en todo instante.

La visión del proyecto es solucionar el problema, brindando dichos servicios de una manera de fácil acceso, es decir vía internet. Dicha problemática es común, lo que constituye una oportunidad de negocio explotable.

4.2.2.2 Planteamiento del problema

Cual es Problema?

La manera deficiente de brindar información económica al alumnado.

Las molestias generales de no tener la información web.

Quienes son los más afectados?

El alumnado, el cual no cuenta con una apropiada manera de obtener información; los administrativos, al no contar con un medio web adecuado de publicación de deudas.

Cual es el impacto del problema?

Al no existir dicho servicio se ocasionan molestias tanto al alumnado, y personal administrativo de la Facultad.

Cual puede ser una solución exitosa?

Contar con un servicio que satisfaga todas las necesidades y que permita la disponibilidad de la información en cualquier momento y lugar, donde sea posible conectarse a internet.

4.2.2.3 Planteamiento de la posición del Servicio

Dirigido a:

Facultades de Universidades

Para Quien:

Facultades con problemas de servicio de información de deudas de alumnos.

El Servicio:

Servicios internet de información económica a nivel Facultad

Diferente de:

Servicio de outsourcing de construcción de soluciones a medida.

Nuestro Servicio ofrece:

Construcción sobre una plataforma más difundida.

Bajos costos al tratarse como un paquete de soluciones.

Funcionamiento para internet.

Adaptabilidad sobre cualquier sistema de pagos y gestión de cobranzas interno que tenga salida a internet.

4.2.3 DESCRIPCIÓN DE STAKEHOLDERS Y USUARIOS

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y

asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

4.2.3.1 Resumen de Stakeholders

Nombre:

Grupo de Servicio de Información Económica

Representa:

Grupo que Evalúa y Dirige el Proyecto

Rol:

Dirigir la construcción de la Solución

Otorgar la información necesaria para el proyecto.

Evaluar los resultados obtenidos.

4.2.3.2 Resumen de Usuarios

Nombre:

FISI

Descripción:

Entidad educativa componente de una institución universitaria.

Stakeholder:

Grupo de Servicio de Información Económica

4.2.3.3 Ambiente de Usuario

El entorno en el cual se desenvuelve el usuario es la web. El uso de la aplicación estará limitado en sus diferentes niveles por el usuario administrativo.

4.2.3.4 Perfil de los Stakeholders

Representante:

Nelson Reyna Vásquez

Descripción:

Grupo de Servicio de Información Económica

Tipo:

Jefe de la Unidad de Economía de la FISI - UNMSM

Responsabilidades:

Encargado de autorizar y aprobar los requerimientos de información. Además, de evaluar los resultados obtenidos del proyecto.

Criterio de Éxito:

Que el servicio colocado inicialmente en la Facultad sea de atracción para otras Facultades y universidades con la misma problemática.

Compromiso:

Para el UML un Business Actor

Grado de participación:

Revisión de requerimientos del sistema.

4.2.3.5 Perfiles de Usuario

Representante

Facultades de nuestra y otras universidades.

Descripción:

Todas aquellas Facultades con problemas de manejo de información

Tipo:

Alumnos y Administrativos familiarizados con el uso de internet.

Responsabilidades:

Hacer uso adecuado del servicio que se presta, respetando los permisos otorgados por el sistema.

Criterio de Éxito:

Encontrar en el servicio una aplicación funcional para la institución que resalte a largo plazo una solución más beneficiosa que optar por un sistema a medida.

Compromiso:

Para el UML 2 actores

Entregables:

Ingreso de datos para activar el sistema

4.2.3.6 Representante Necesidades de los Stakeholders y los Usuarios

Una de las necesidades de los stakeholders, para poder hacer viable el proyecto es el conocimiento de las diversas estructuras de las Base de datos usadas para los futuros usuarios, esto constituye un obstáculo, es por esto que se ha optado por una BD independiente.

4.2.3.7 Alternativas y Competencia

Existen muchas empresas que brindan servicios de desarrollo de soluciones a medida pero son de elevado costo, lo cual no constituye una empresa significativa.

Un factor que se puede considerar como competencia, es que otras Facultades en otras universidades estén en condiciones a desarrollar el mismo proyecto al presentarles solución.

4.2.4 DESCRIPCIÓN GLOBAL DEL SERVICIO

4.2.4.1 Perspectiva del Servicio

Nuestra intención es brindar servicio de información económica, que soluciona, que soluciona una problemática específica de publicación de la información.

Inicialmente esta solución se orienta al uso dentro de la misma Facultad, tomada como prueba piloto, para luego orientarse a demás Facultades tanto de nuestra universidad como de otras. La implementación inicial de la solución permitirá tener una muestra que respalde la eficiencia del servicio, con la prueba del servicio en un entorno que cuente con más usuarios.

4.2.4.2 Resumen de capacidades

A continuación se mostrará un listado con los beneficios que se obtendrá del servicio:

Uso de internet e independencia de plataforma del cliente.

Seguridad y Privacidad mediante de uso de autenticación

Servicio diferenciado para los administrativos y alumnos.

4.2.4.3 Suposiciones y dependencias

El usuario es considerado la suficiente responsable en el uso de las opciones que le corresponden, el sistema depende de la conexión de internet.

4.2.4.5 Costo y precio

Los costos de la solución, se basara en reducir en costos similares de implementación de la solución por outsourcing, para competir de manera eficiente.

4.3 ESPECIFICACIONES SUPLEMENTARIAS

4.3.1 INTRODUCCIÓN

4.3.1.1 Propósito

El propósito de este documento es definir los requerimientos que no se especificarán en la documentación RUP para los casos de uso, también se incluyen características que pueden limitar este sistema como usabilidad, disponibilidad y desempeño.

4.3.1.2 Alcance

Se describen brevemente la funcionalidad que no está especificada en los casos de uso, completando todos los requerimientos para la construcción del sistema. También se describen las características que establecen la eficiencia y potencialidad que dan a los sistemas de software.

4.3.1.3 Definiciones, Acrónimos, y Abreviaciones

Ver Glosario al final del Documento

4.3.2 USABILIDAD

Describe las características que apuntan al usuario final.

4.3.2.1 Facilidad de uso

El sistema será desarrollado para el manejo fácil, rápido e intuitivo para el usuario enmarcado directamente en las rutinas diarias de cada uno.

4.3.2.2 Capacitación

La Capacitación podrá realizarse según el tipo de rol que tenga el usuario, llevando no más de 1 hora por cada rol.

4.3.2.3 Interfaces de Usuario

El diseño del sistema tendrá una presentación sencilla y amigable al usuario.

4.3.2.4 Disponibilidad

El Sistema podrá estar disponible las 24 horas del día, los 7 días de la semana.

4.3.2.5 Desempeño

4.3.2.5.1 Tiempo de respuesta

El tiempo de respuesta del sistema, dependerá de la velocidad de conexión, debido a la interacción con la Base de datos, se considera un valor promedio para el sistema el tiempo de respuesta de 1 a 2 segundos.

4.3.2.5.2 Carga de Trabajo

El tiempo de respuesta del sistema está limitado a la Oficina de Economía de la FISI, el número de usuarios será los empleados de la misma, sin embargo las consultas web de los alumnos aumentará el número accesos realizadas al motor de Base de Datos.

4.3.2.5.3 Recursos

El sistema podrá funcionar apropiadamente con las siguientes características mínimas:

Servidor Web: disco duro de 50 GB, memoria RAM de 512 MB, procesador de 1.5 GHz, tarjeta de red 10/100 Mbps, velocidad de conexión a Internet con una tasa de transferencia de datos de 128 Kbps.

Servidor de Base de Datos: disco duro de 50 GB, memoria RAM de 512 MB, procesador de 1.5 GHz, tarjeta de red 10/100 Mbps, velocidad de conexión a Internet con una tasa de transferencia de datos de 128 Kbps.

4.4 ANÁLISIS

El análisis del sistema se lo realiza a través de los casos de uso. Los casos de uso del sistema encierran la funcionalidad total del sistema y son una representación de alto nivel de los requerimientos del negocio.

4.4.1 MODELOS DE CASOS DE USO GENERAL DEL SISTEMA

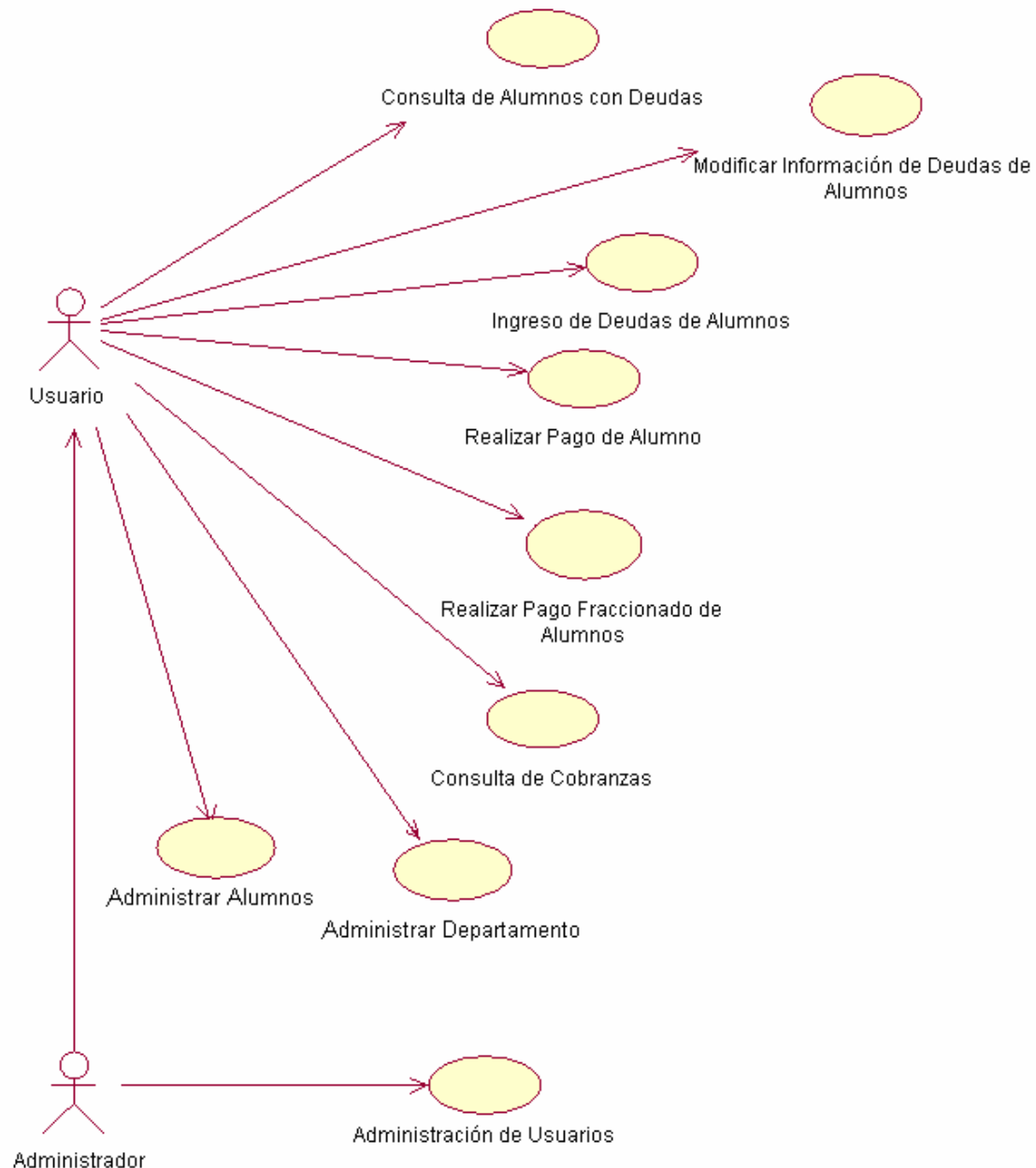


Figura 4.1: Modelo de Casos de Uso General

4.4.2 LISTA DE CASOS DE USO

Ingreso de Deudas de Alumnos

Realizar Pago de Alumno

Realizar Pago Fraccionado de Alumnos

Modificación información de Deudas

Reporte de Cobranzas

Reporte de Deudas

Administración de Usuarios

Se describen las especificaciones de los casos de uso como Ingreso de Deudas de Alumnos, Realizar Pago de Alumno, Pago Fraccionado de Pago de Alumnos, Modificar Información de deudas de Alumnos, Reporte de Deudas, Reporte de Cobranzas, Administrar de usuarios.

4.4.2.1 Caso de uso Ingreso de Deudas de Alumnos

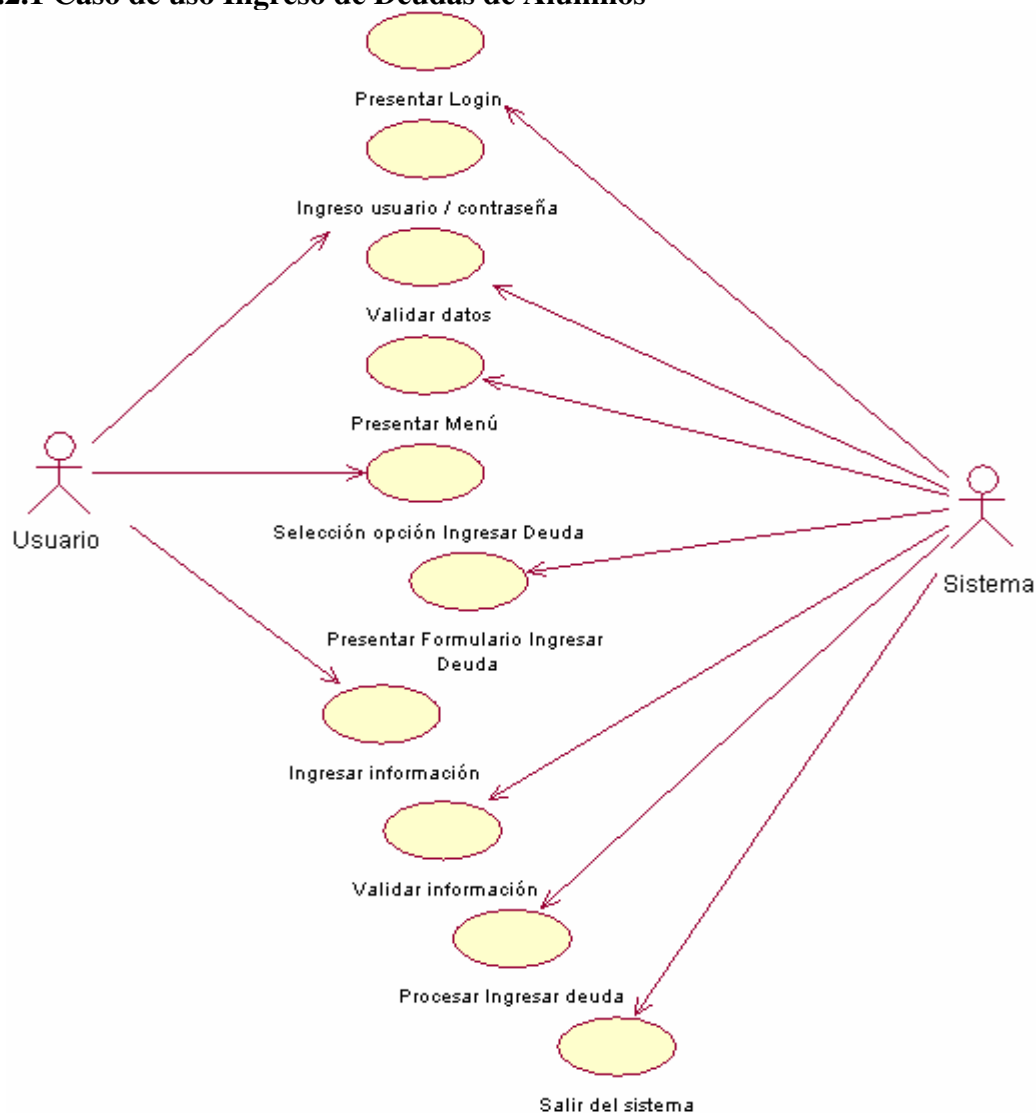


Figura 4.2: Modelo de Caso de Uso Ingreso de Deudas de Alumnos

4.4.2.2 Caso de uso Realizar Pago de Alumno

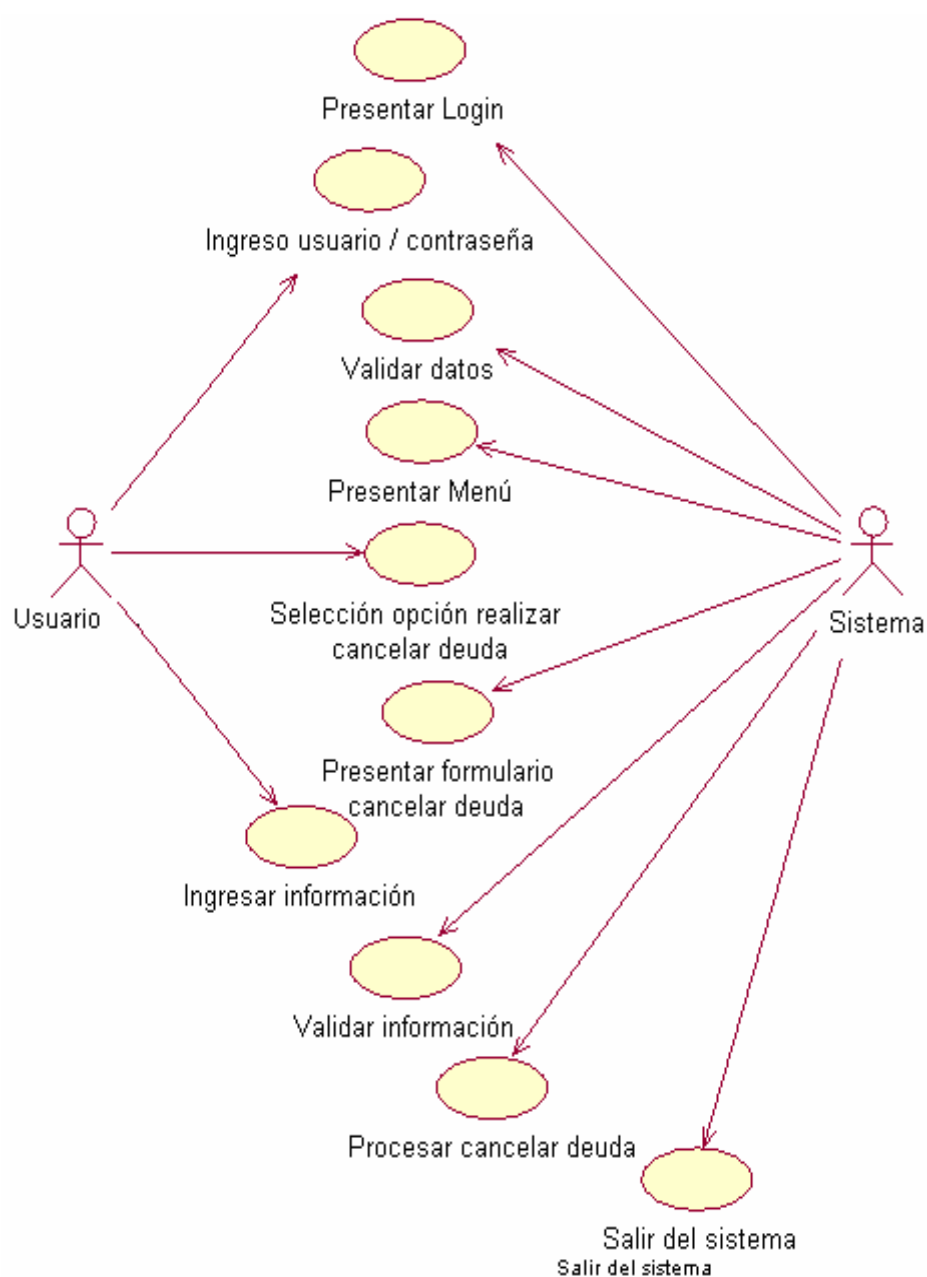


Figura 4.3: Modelo de Caso de Uso Realizar Pago de Alumno

4.4.2.3 Caso de uso Modificar Información de Deudas

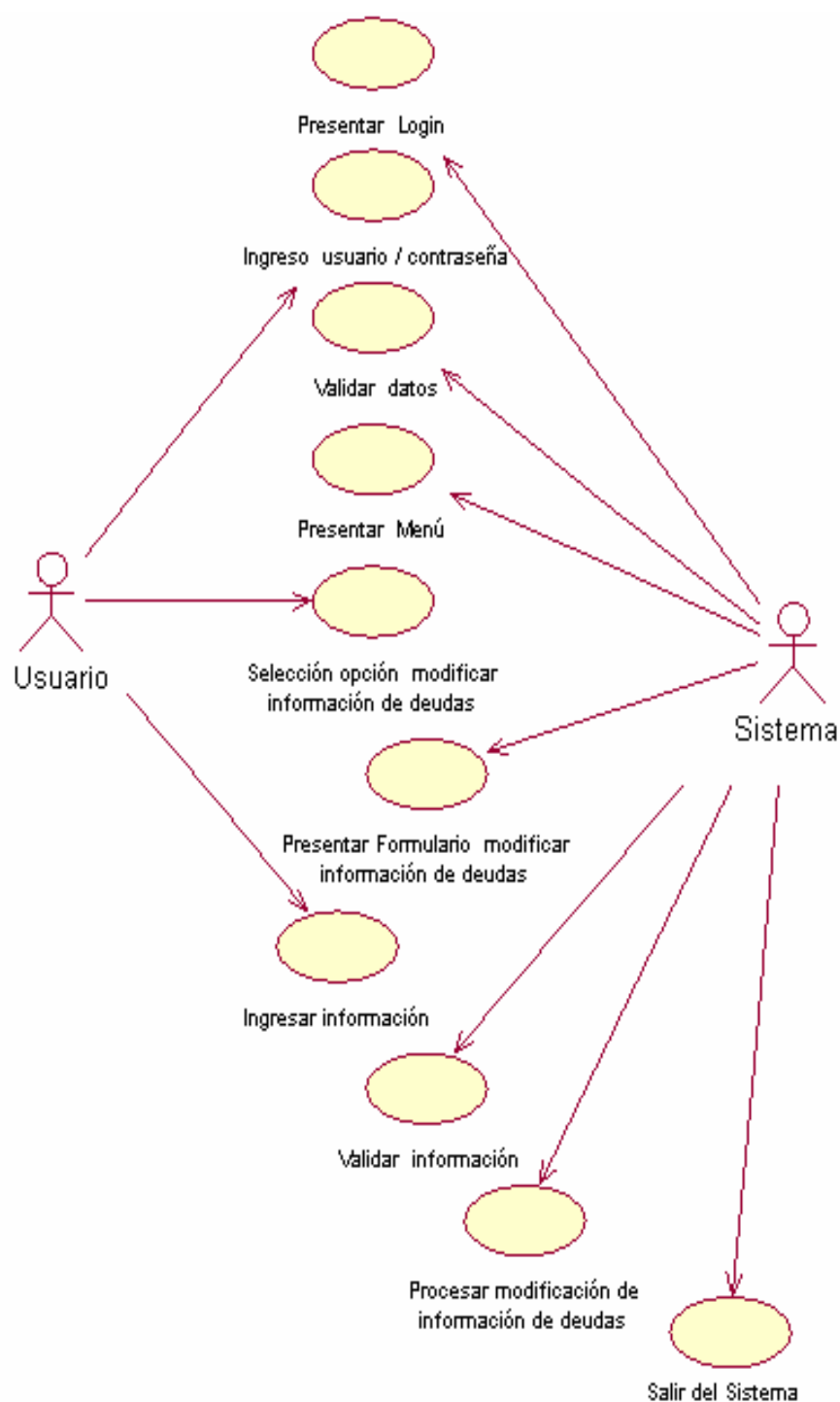


Figura 4.5: Modelo de Caso de Uso Modificar Información de Deudas

4.4.2.4 Caso de uso Realizar Pago Fraccionado de Alumnos

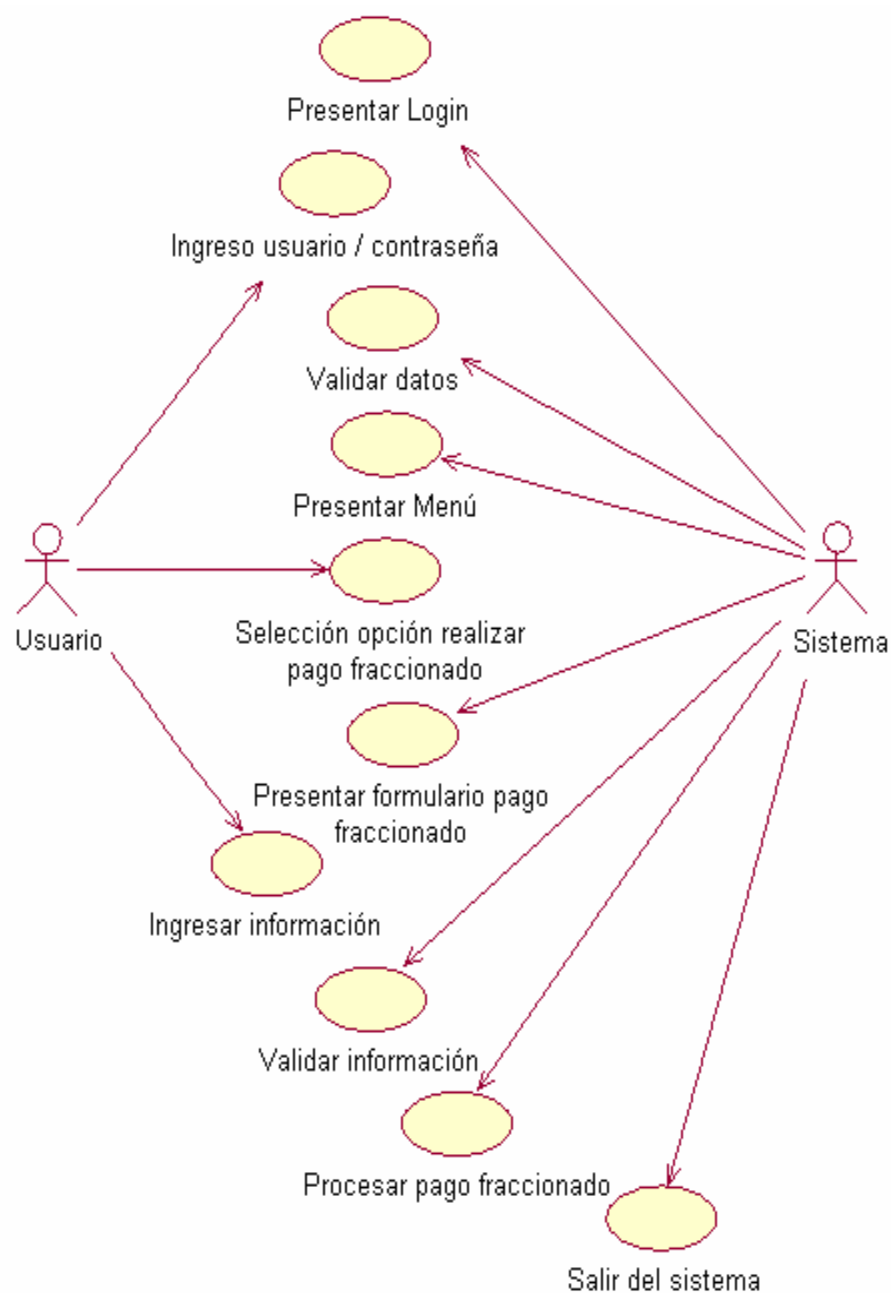


Figura 4.5: Modelo de Caso de Uso Realizar Pago Fraccionado de Alumnos

4.4.2.5 Caso de uso Reporte de Deudas

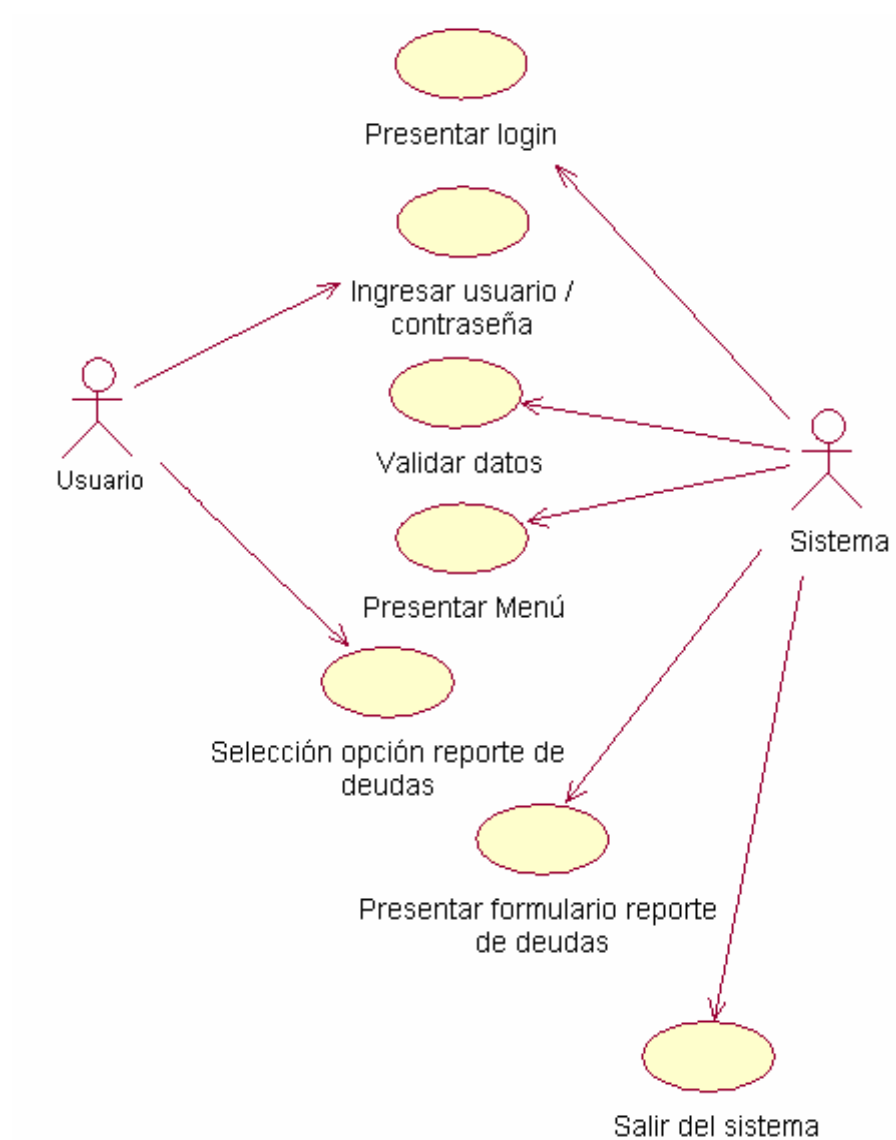


Figura 4.6: Modelo de Caso de Uso Reporte de Deudas

4.4.2.6 Caso de uso Reporte de Cobranzas

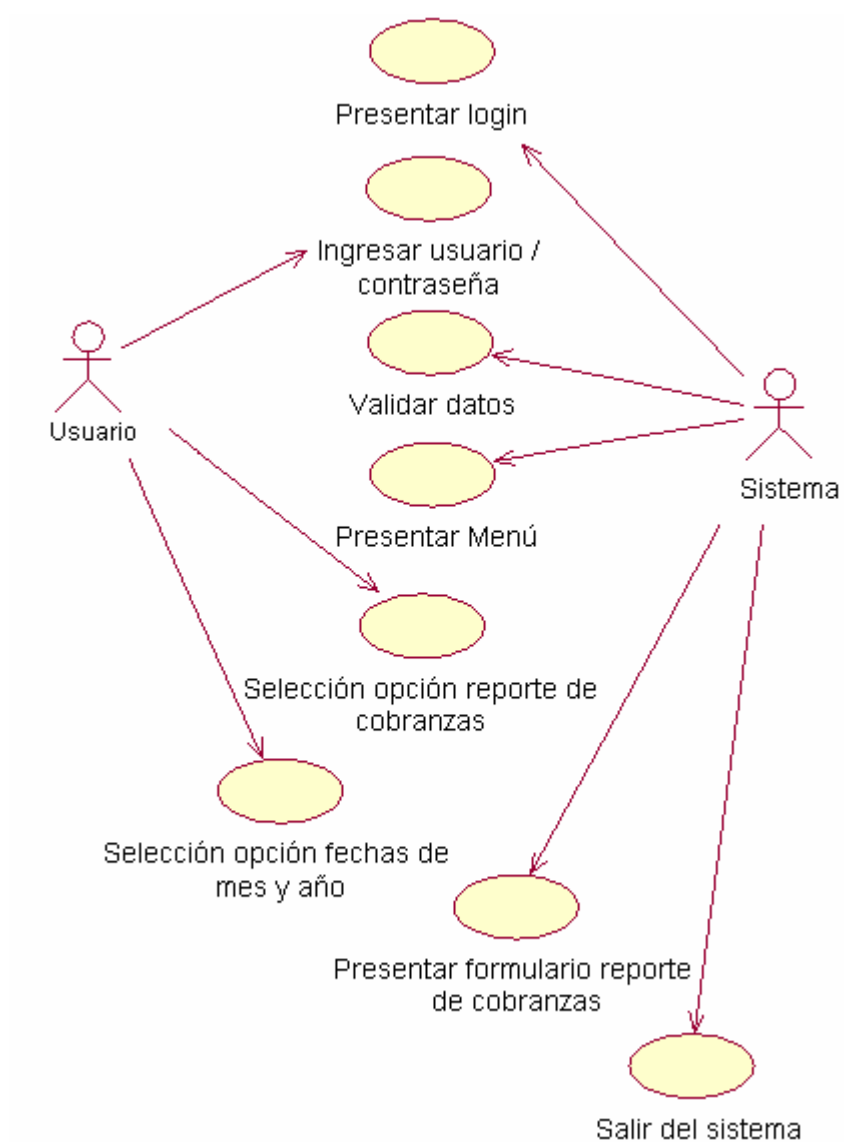


Figura 4.7: Modelo de Caso de Uso Reporte de Cobranzas

4.4.2.7 Caso de uso Administración de Usuarios

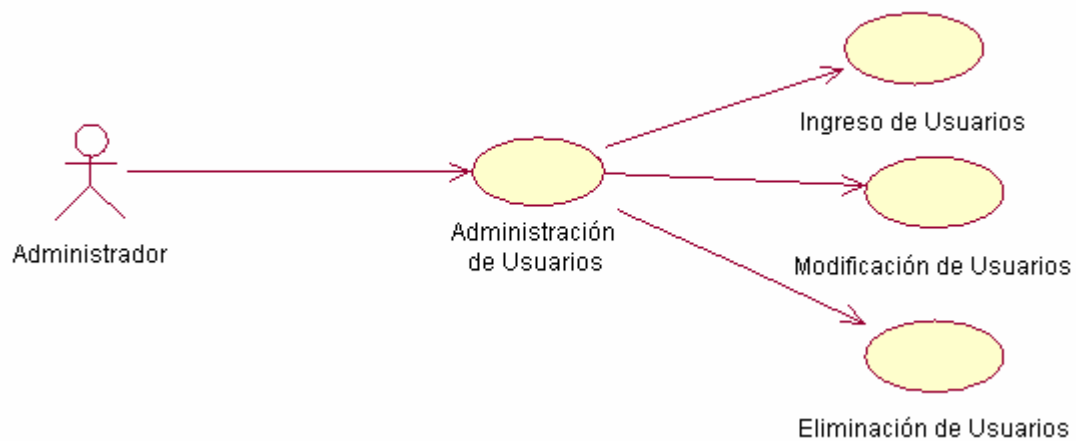


Figura 4.8: Modelo de Caso de Uso Administración de Usuarios

4.4.2.8 Caso de uso Ingreso de Usuarios

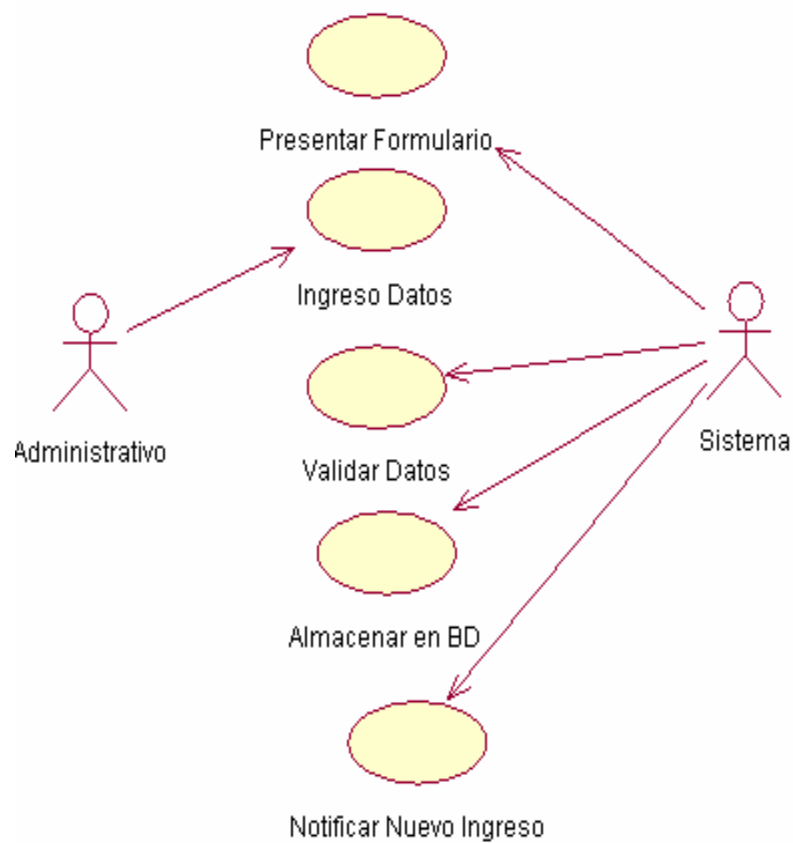


Figura 4.9: Modelo de Caso de Uso Ingreso de Usuarios

4.4.2.9 Caso de uso Modificar datos de Usuarios

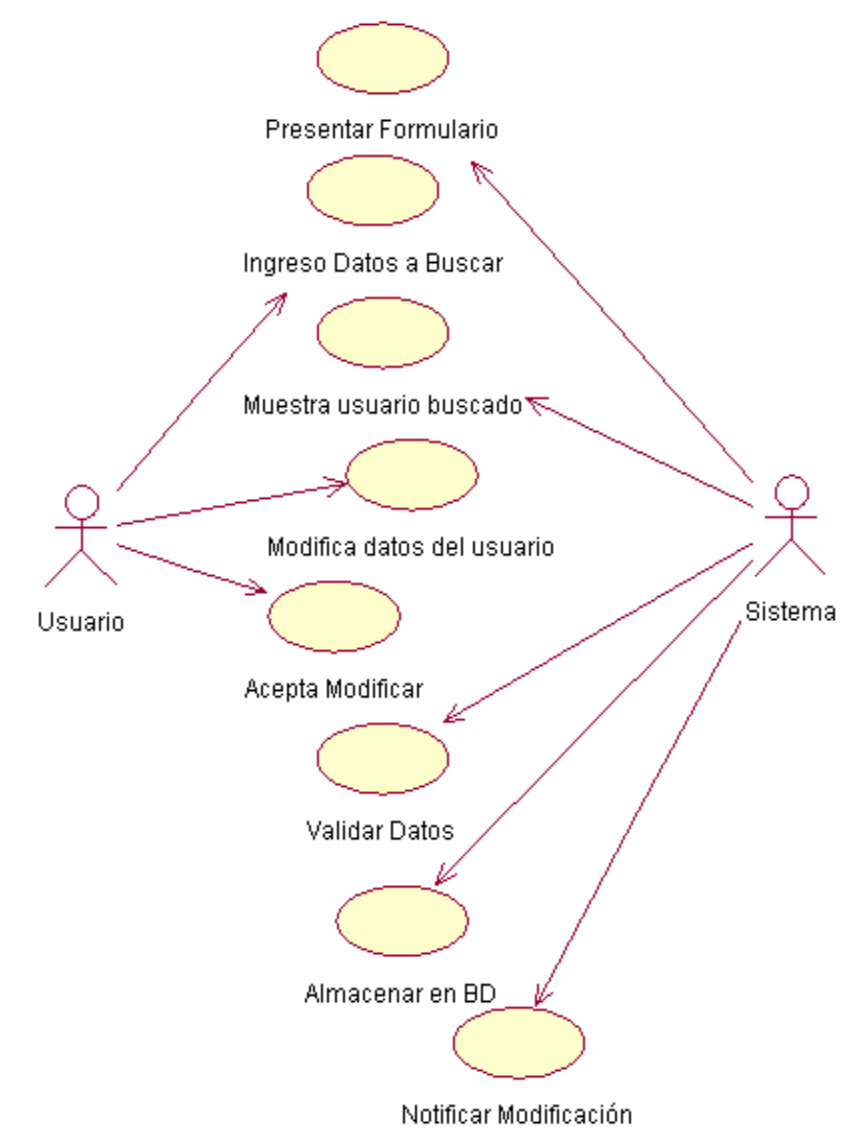


Figura 4.10: Modelo de Caso de Uso Modificar datos de Usuarios

4.4.2.10 Caso de uso Eliminar Usuarios

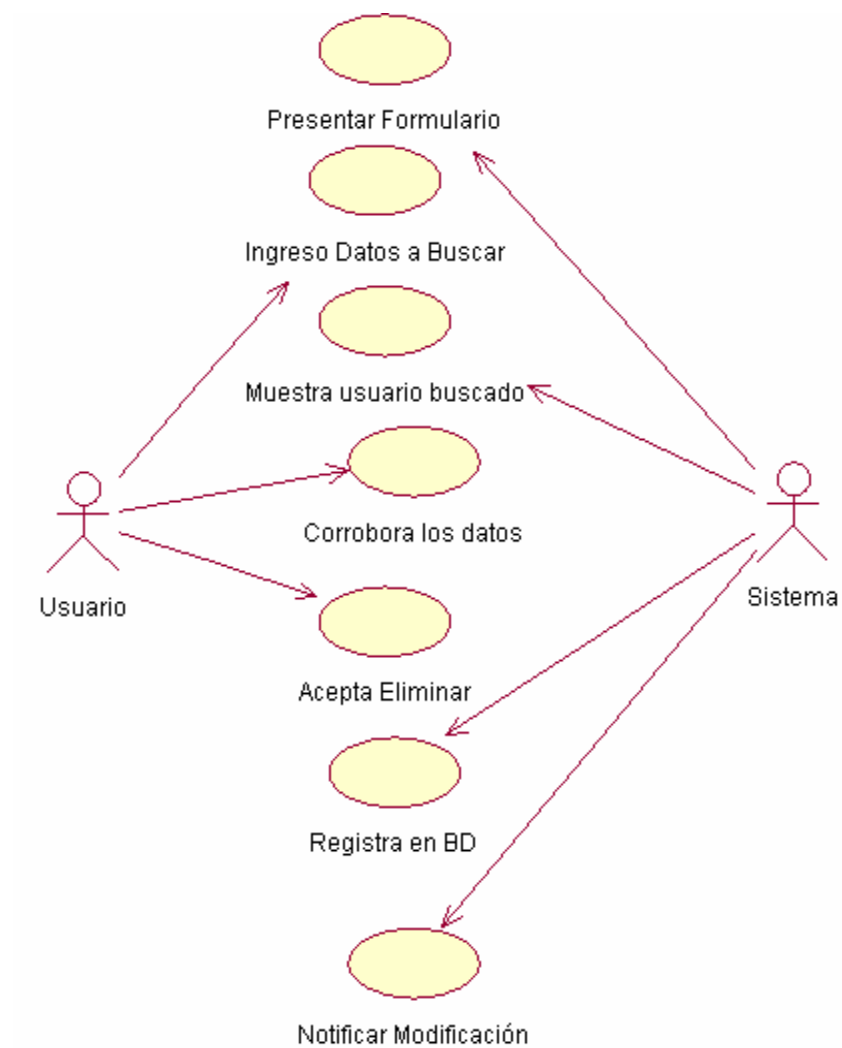


Figura 4.11: Modelo de Caso de Uso Eliminar Usuarios

4.4.2.11 Caso de uso del Sistema

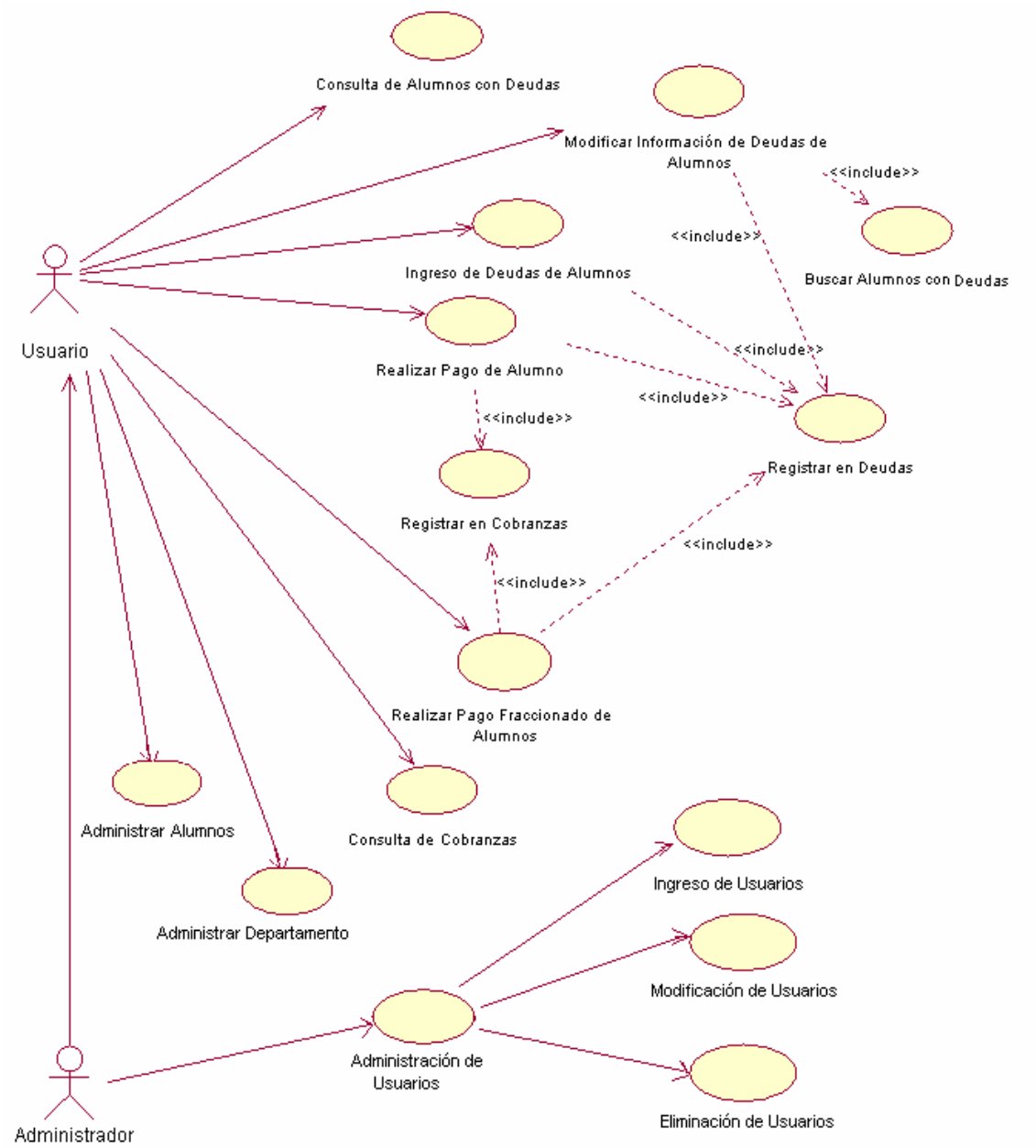


Figura 4.12: Modelo de Casos de Uso del Sistema

4.4.3 ESPECIFICACIÓN DE CASOS DE USO

4.4.3.1 Ingreso de Deudas de Alumnos

Descripción

Describe el proceso para el ingreso correspondiente a la deuda de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Ingreso de Deudas	
	2. Muestra en pantalla el formulario para cancelar el pago de las deudas pendientes de alumnos.
3. Busca los datos del alumno.	
	4. Muestra pantalla para los ingresos de datos como fecha de origen de deuda, monto, etc.
5. Ingresa datos del alumno como su estado, monto de la deuda del alumno y datos de necesarios de la oficina.	
	6. Se registra deuda pendiente del alumno, se agrega al registro de deudas como pendientes.
	7. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para ingresar deuda de los alumnos

El usuario es empleado de la Oficina de Economía.

Poscondiciones

El usuario se encuentra registrado y el monto correspondiente de la deuda del alumno.

4.4.3.2 Realizar Pago de Alumno

Descripción

Describe el proceso para realizar el Pago correspondiente a la deuda de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Pagos.	
	2. Muestra en pantalla el formulario para cancelar el pago de las deudas pendientes de alumnos.
3. Busca los datos del alumno.	
	4. Encuentra la búsqueda y muestra la pantalla de los datos del alumno, como el monto de la deuda.
5. Ingresar datos como la fecha de pago, el número de Voucher, y el registro de Ingreso de Tesorería, etc.	
6. Verifica que los datos sean correctos y acepta el pago total de la deuda.	
	6. Cancela deuda pendiente, se agrega al registro de cobranzas y se elimina en la lista de deudas pendientes.
	7. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para realizar pagos de los alumnos
El usuario es empleado de la oficina de economía.

Poscondiciones

El usuario se encuentra registrado, también ha cancelado el monto correspondiente de la deuda de alumnos y se registro en Cobranzas.

4.4.3.3 Realizar Pago Fraccionado de Alumnos

Descripción

Describe el proceso para realizar el Cobro fraccionado correspondiente a la deuda de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Pago Fraccionado.	
	2. Muestra en pantalla el formulario para pagar una fracción del pago de las deudas pendiente del alumno.
3. Busca los datos del alumno.	
	4. Encuentra la búsqueda y muestra la pantalla de los datos del alumno, como el monto de la deuda.
5. El ingreso de datos, ingresa la fracción de la deuda del alumno, datos como la fecha de pago, el número de Voucher, el recibo de Ingreso de Tesorería, etc.	
6. Verifica que los datos sean correctos y confirma la fracción del pago.	
	7. Cancela una fracción de deuda pendiente y se agrega al registro de cobranzas y se actualiza en la lista de deudas pendientes.
	8. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para realizar los Cobros fraccionados de los alumnos
El usuario es empleado de la oficina de economía.

Poscondiciones

El usuario se encuentra registrado, también ha cancelado una fracción correspondiente de la deuda de alumnos y se registro en las Cobranzas.

4.4.3.4 Reporte de Cobranzas

Descripción

Describe el proceso para realizar las Consultas de Cobranzas de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Reporte de Cobranzas.	
	2. Muestra en pantalla el menú para las consultas de los pagos hechos por mes y año.
3. Busca los registros de los alumnos.	
	4. Encuentra los datos y muestra la pantalla de los alumnos, mostrando el monto acumulado en soles de los pagos y su fecha.
6. Verifica que los datos sean correctos.	
	7. Cierra la Consulta
	8. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para Consultar de las deudas de los alumnos

El usuario es empleado de la oficina de economía.

Poscondiciones

El usuario se encuentra registrado, consulto el registro de pagos realizados por los alumnos en las Cobranzas.

4.4.3.5 Reporte de Deudas

Descripción

Describe el proceso para modificar la información de deudas de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Reportes de Deudas.	
	2. Muestra en pantalla el formulario para las consultas de las deudas de los alumnos.
3. Busca los registros de los alumnos.	
	4. Encuentra los datos y muestra la pantalla de los alumnos, mostrando el monto en soles de las deudas y fecha.
6. Verifica que los datos sean correctos.	
	7. Cierra Reporte.
	8. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para Consultar de las deudas de los alumnos

El usuario es empleado de la oficina de economía.

Poscondiciones

El usuario se encuentra registrado, consulto el registro de los alumnos con deudas.

4.4.3.6 Modificar Información de Deudas de Alumnos

Descripción

Describe el proceso para modificar la información de deudas de los alumnos.

Flujo de Eventos

Flujo Básico

Eventos ACTOR	Eventos SISTEMA
1. Activa función de Modificar información de Deudas de Alumnos.	
	2. Muestra en pantalla el formulario para la modificar la información de deudas de los alumnos.
3. Busca los registros de los alumnos.	
	4. Encuentra los datos y muestra en la pantalla los datos referentes a la deuda del alumno.
6. Verifica que los datos sean correctos.	
7. Modifica los valores referentes a la deuda del alumno y graba los cambios.	
	8. Cierra la Consulta
	9. Reinicia el CU.

Precondiciones

El usuario ha sido habilitado para Consultar de las deudas de los alumnos

El usuario es empleado de la oficina de economía.

Poscondiciones

El usuario se encuentra registrado, consulto el registro de pagos realizados por los alumnos en las Cobranzas.

4.4.3.7 Administrar Usuarios

Descripción

Administra los usuarios del sistema.

Flujo de Eventos

Flujo Básico Ingresar:

Eventos ACTOR	Eventos SISTEMA
	1. Muestra en pantalla el formulario para el ingreso de nuevos usuarios.
2. Ingresar datos del usuario (nombre, password) y hace click en “Agregar”.	
	3. Verifica que los datos sean correctos.

Flujo Básico Modificar:

Eventos ACTOR	Eventos SISTEMA
	1. Muestra en pantalla el formulario para la modificación de usuarios.
2. Ingresar datos del usuario a buscar.	
	3. Muestra los datos del usuario encontrado.
4. Modifica los datos del usuario y hace click en “Modificar”.	
	5. Verifica que los datos sean correctos.

Flujo Básico Eliminar:

Eventos ACTOR	Eventos SISTEMA
	1. Muestra en pantalla el formulario para la eliminación de usuarios.
2. Ingresar datos del usuario a buscar.	
	3. Muestra los datos del usuario encontrado.
4. Corroborar los datos y hace click en “Eliminar”.	
	5. Verifica los datos y actualiza la BD.

Precondiciones

Usuarios del sistema cargados.

Poscondiciones

Usuarios actualizados.

4.5 DISEÑO

4.5.1 ARQUITECTURA DEL SISTEMA

4.5.1.1 Introducción

4.5.1.1.1 Propósito

Este documento describe los aspectos estáticos y dinámicos del sistema utilizando varias vistas de la arquitectura para describir diferentes aspectos del sistema, identificando el orden de construcción del sistema que es guiado por los casos de uso y la relación existente entre todos los involucrados, dando una perspectiva clara del sistema.

4.5.1.1.2 Alcance

Este documento contiene las siguientes vistas: vista de Casos de Uso, Lógica, de Datos, Despliegue e Implementación, las que describen la arquitectura total del sistema.

4.5.1.1.3 Definiciones, Acrónimos y Abreviaturas

Ver Glosario al final del Documento

4.5.1.1.4 Referencias

Modelos de Casos de uso

4.5.1.2 Representación de la Arquitectura

Este documento describe la arquitectura a través de varios tipos de vistas enfocándose en diferentes aspectos, con el propósito de comunicar de la mejor manera los componentes del sistema.

4.5.1.2.1 Vista de Casos de Uso

Describe un juego de caso de uso que representan situaciones significativas, está centrado en la funcionalidad del sistema.

4.5.1.2.2 Vista Lógica

Esta vista documenta el modelo de diseño, el cual define las capas de la aplicación y las clases primarias en cada capa, la organización en paquetes y subsistemas, y la organización de estos paquetes por capas.

4.5.1.2.3 Vista de Despliegue

Describe la topología física del sistema, describe cómo los nodos se comunican, especificándose la configuración para cada uno.

4.5.1.2.4 Vista de Implementación

Describe las capas de aplicación.

4.5.1.3 Vista de Casos de Uso

Se describen brevemente los casos de uso del sistema.

- **Ingreso de Deudas de Alumnos:** Permite el ingreso de los alumnos que se registrarán en el sistema
- **Realizar Pago de Alumno:** El sistema realizará cancelando la deuda del alumno eliminando de las deudas y registrando en las Cobranzas.
- **Realizar Pago Fraccionado de Alumnos:** Realizará el pago fraccionado en el sistema cancelando la deuda parcial del alumno disminuyendo el monto en la lista de deudas y registrando en las Cobranzas.
- **Consulta de Cobranzas:** Realizará la consulta por fechas de cobros realizados a los alumnos mostrando sus montos.
- **Consulta de Alumnos con Deudas:** Realizará la consulta de las deudas de alumnos mostrando sus montos.
- **Modificar Información de Deudas:** Modificará la información de las deudas actualizar los datos del estado del alumno.
- **Administración de Usuarios:** Ingresará nuevos, modificará y eliminará usuarios.

4.5.1.3.1 Modelos de Actores



Figura 4.13: Modelo de Actores

Actor	Descripción	Usuarios
Administrador	Representa al usuario encargado de la administración del sistema.	Administrador del Sistema
Usuario	Cualquier usuario del sistema que puede ser asignado que pertenece como empleado del área administrativa.	Usuarios administrativos

4.5.1.4 Vista Lógica

4.5.1.4.1 Vista General

Esta vista representa el núcleo del diseño del sistema, presente a las clases primarias que colaboran con la implementación de la funcionalidad del sistema, para lo cual se describe la organización en paquetes distribuidos en diferentes capas, los cuales son:

- **Paquete de Vistas:** Contiene todas las vistas para la interacción con el usuario final. Es un paquete de la capa de Presentación.
- **Paquete de Modelos:** Contiene clases para importación y almacenamiento del esquema de base de datos. Es un paquete de la capa de Acceso a Datos.
- **Paquete de Controles:** Contiene las clases que controlan la funcionalidad total del sistema, es un paquete de la capa del Negocio.
- **Paquete de Dominio:** Contiene los paquetes que agrupan a las clases del sistema.

Paquete de Vistas:

Por la extensión existente en el número de vistas en el paquete de vistas, se subdividen de acuerdo a cada caso de uso.

Ingreso de Deudas de Alumnos



Figura 4.14: Inteface Ingreso de Deudas de Alumnos

Realizar Pago de Alumno

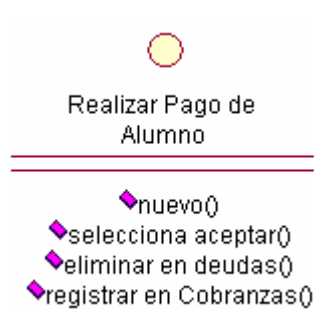


Figura 4.15: Inteface Realizar Pago de Alumno

Realizar Pago Fraccionado de Alumnos

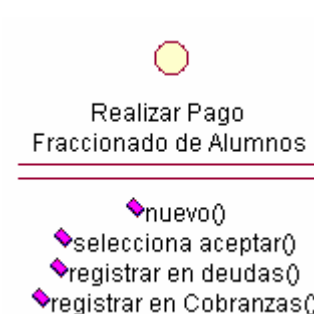


Figura 4.16: Inteface Realizar Pago Fraccionado de Alumnos

Consulta de Cobranzas

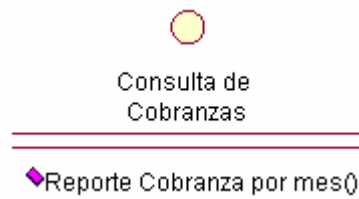


Figura 4.17: Inteface Consulta de Cobranzas

Consulta de Alumnos con Deudas

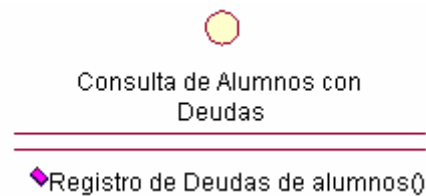


Figura 4.18: Inteface Consulta de Alumnos con Deudas

Paquete de Controles:

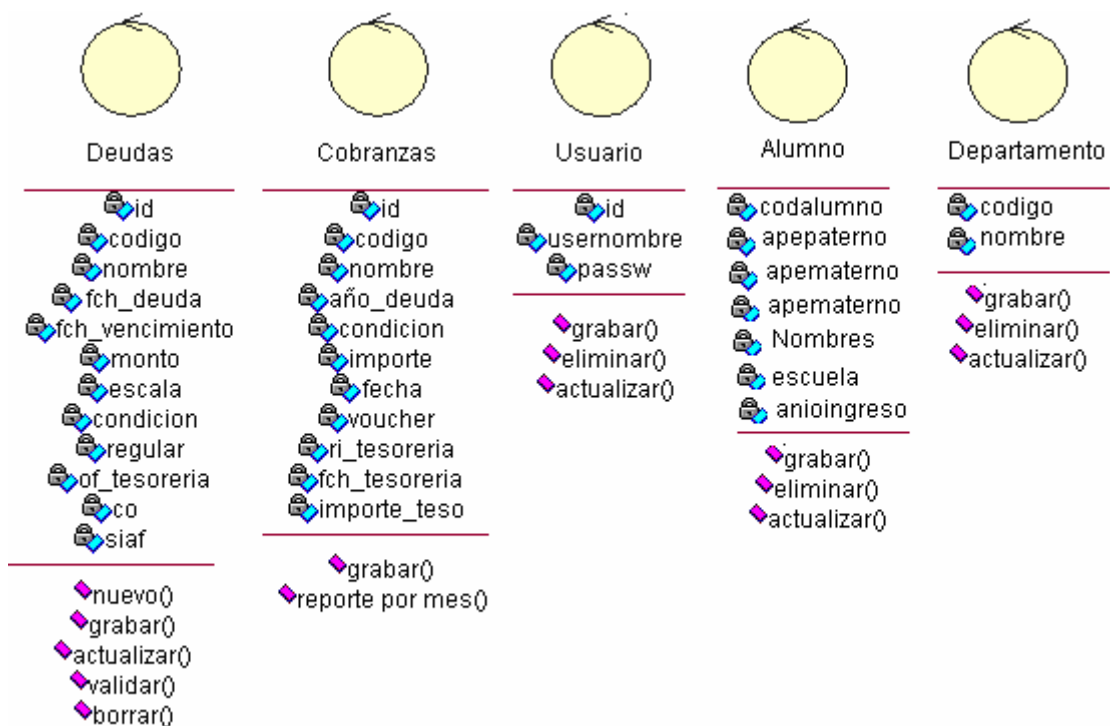


Figura 4.19: Clases que manejan la Lógica del Negocio

Paquete de Modelos:

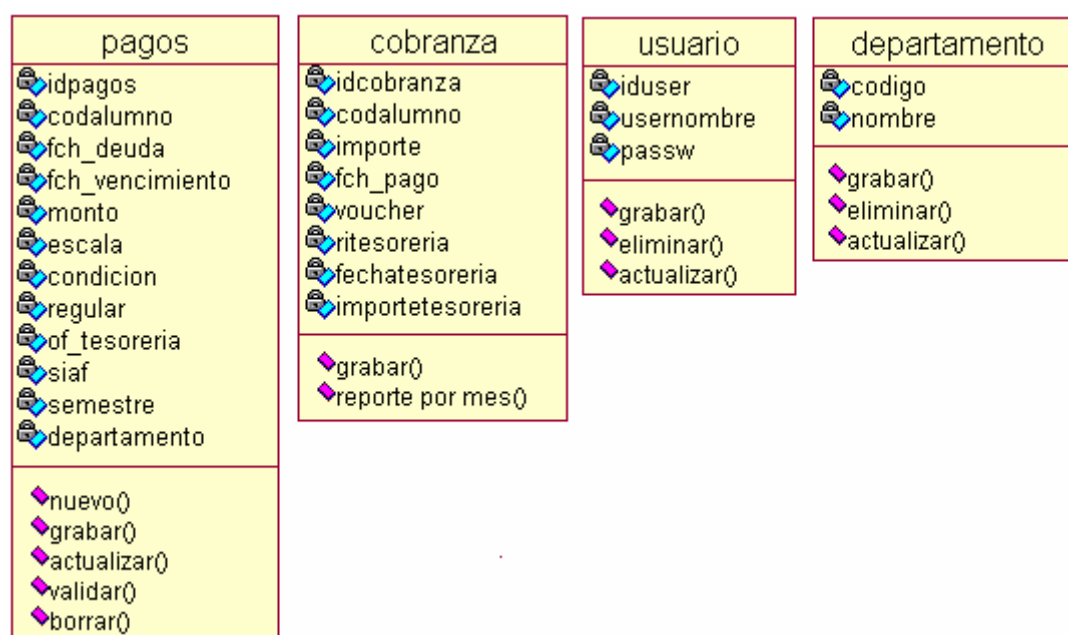


Figura 4.20: Diagrama de Clases

Paquete de Dominio:

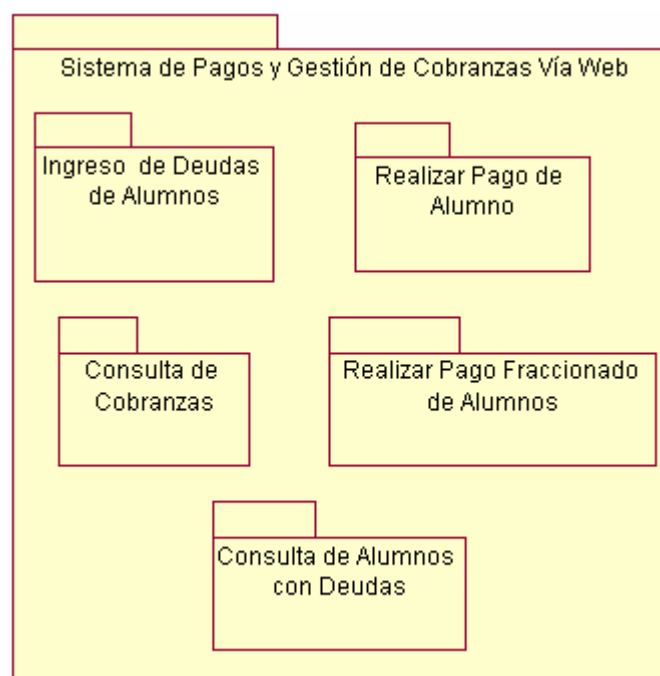


Figura 4.21: Paquete de Dominio

4.5.1.5 Vista de Despliegue

A continuación se presenta el modelo que contiene cada uno de los componentes que se integran sobre el sistema.



Figura 4.22: Diagrama de Despliegue

4.5.1.5.1 Servidor Web

Software	Versión	Vendedor	Comentarios
ASP. Net	2005	Microsoft	

Cuadro 4.5.1: Servidor Web

4.5.1.5.2 Servidor Base de Datos

Software	Versión	Vendedor	Comentarios
MySQL	5.0	Licencia Libre	Comunicación IP habilitada

Cuadro 4.5.2: Servidor de Base de Datos

4.5.1.5.3 Estaciones Cliente

Software	Versión	Vendedor	Comentarios
Internet Explorer	5 o superior	Licencia Libre	

Cuadro 4.5.3: Estaciones Cliente

4.5.1.6 Vista de Implementación

4.5.1.6.1 Relación entre Capas y Paquetes

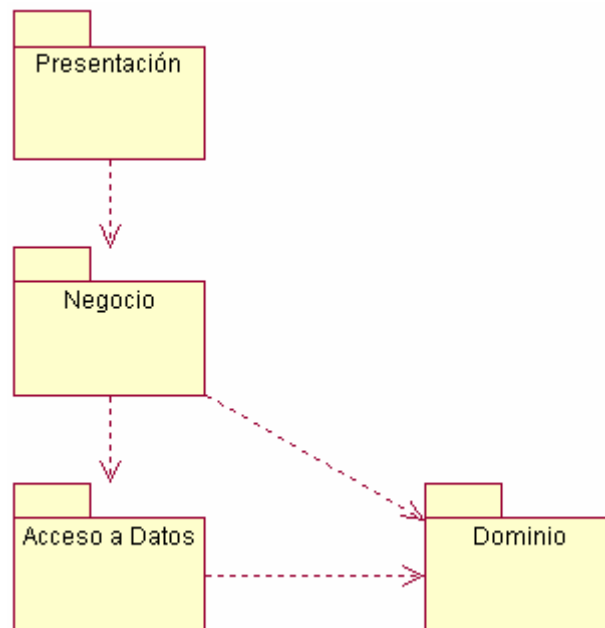


Figura 4.23: Capas del Sistema

4.5.1.6.2 Capas

Capa de Presentación: La capa de presentación tiene componentes necesarios para permitir interacción con el usuario final, contiene todas las interfaces de usuario y depende de los objetos del negocio.

Capa de Negocio: La capa del negocio contiene componentes relacionados con la lógica del negocio. Contiene todas las clases que manejan el negocio y dependen de los objetos del negocio.

Capa de Acceso a Datos: La Capa de Acceso a Datos contiene componentes necesarios para habilitar la comunicación entre la capa de aplicación y la información contenida en una base de datos. Contiene todas las clases de entidad que representan tablas en el dominio de la Aplicación.

4.5.1.7 Vista de Componentes

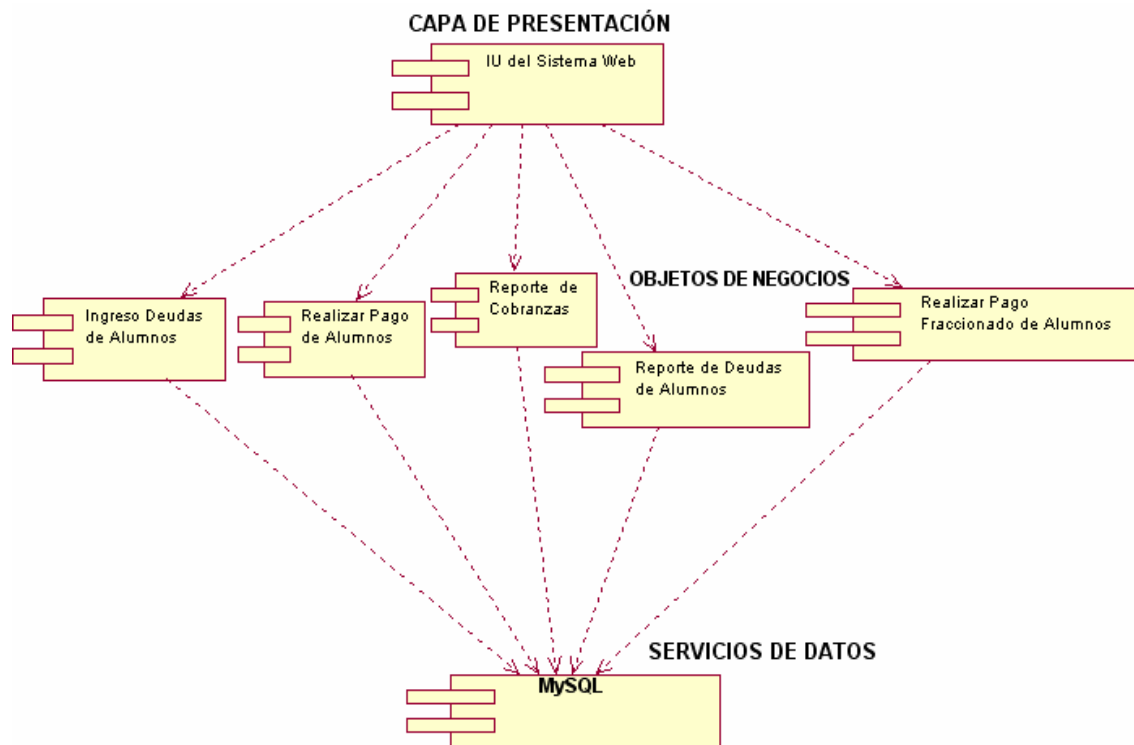


Figura 4.24: Diagrama de Componentes

4.5.1.8 Vista de Colaboración

4.5.1.8.1 Realizar Pago de Alumnos

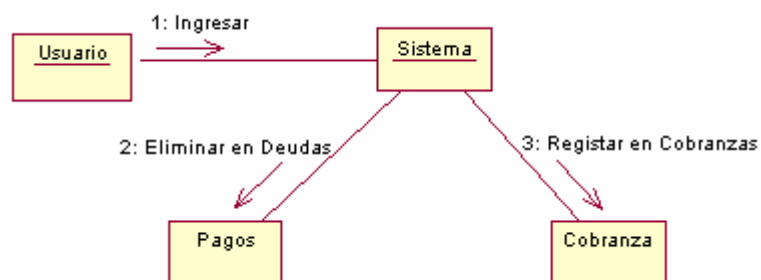


Figura 4.25: Diagrama de Colaboración - Realizar Pago de Alumnos

4.5.1.8.2 Realizar Pago Fraccionado de Alumnos

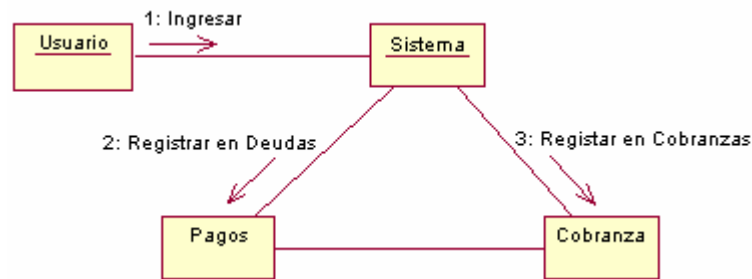


Figura 4.26: Diagrama de Colaboración - Realizar Pago Fraccionado de Alumnos

4.5.2 REALIZACIÓN DE CASOS DE USO

La realización de los casos de uso más significativos se la representa mediante los diagramas de actividad, secuencia, comunicación e interfaces de usuario.

4.5.2.1 Realizar Pago de Alumno

4.5.2.1.1 Diagrama de Estados

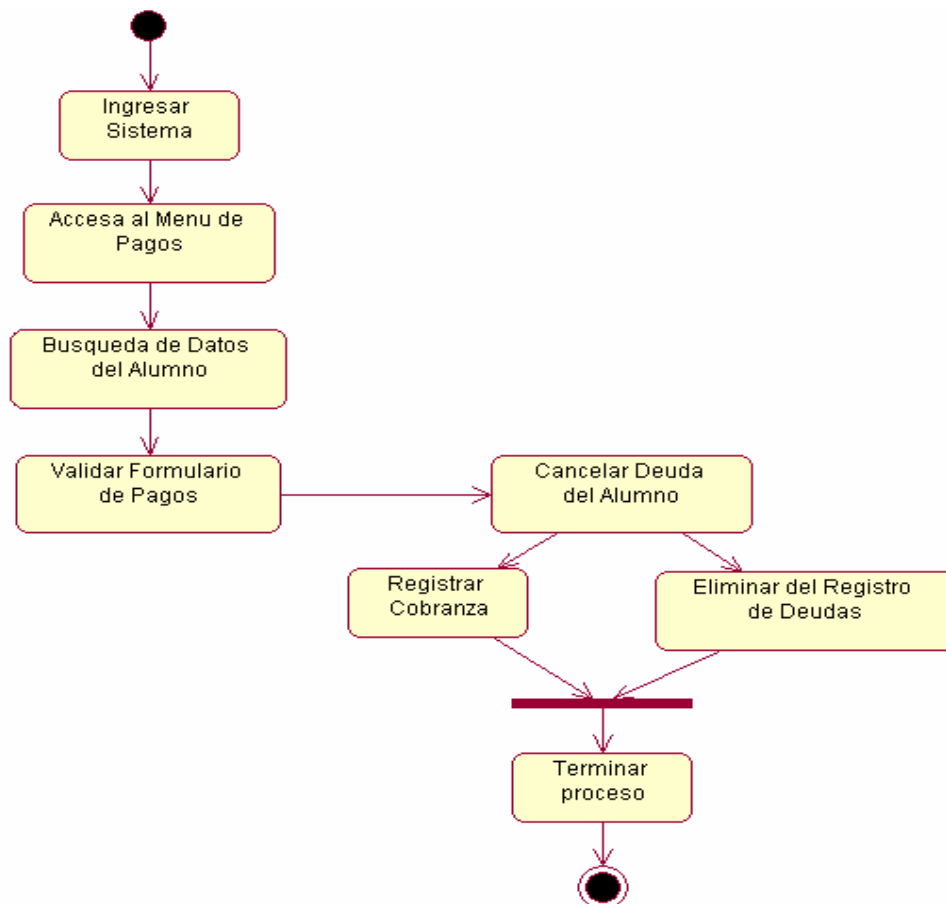


Figura 4.27: Diagrama de Estados - Realizar Pago de Alumno

4.5.2.1.2 Diagrama de Secuencias

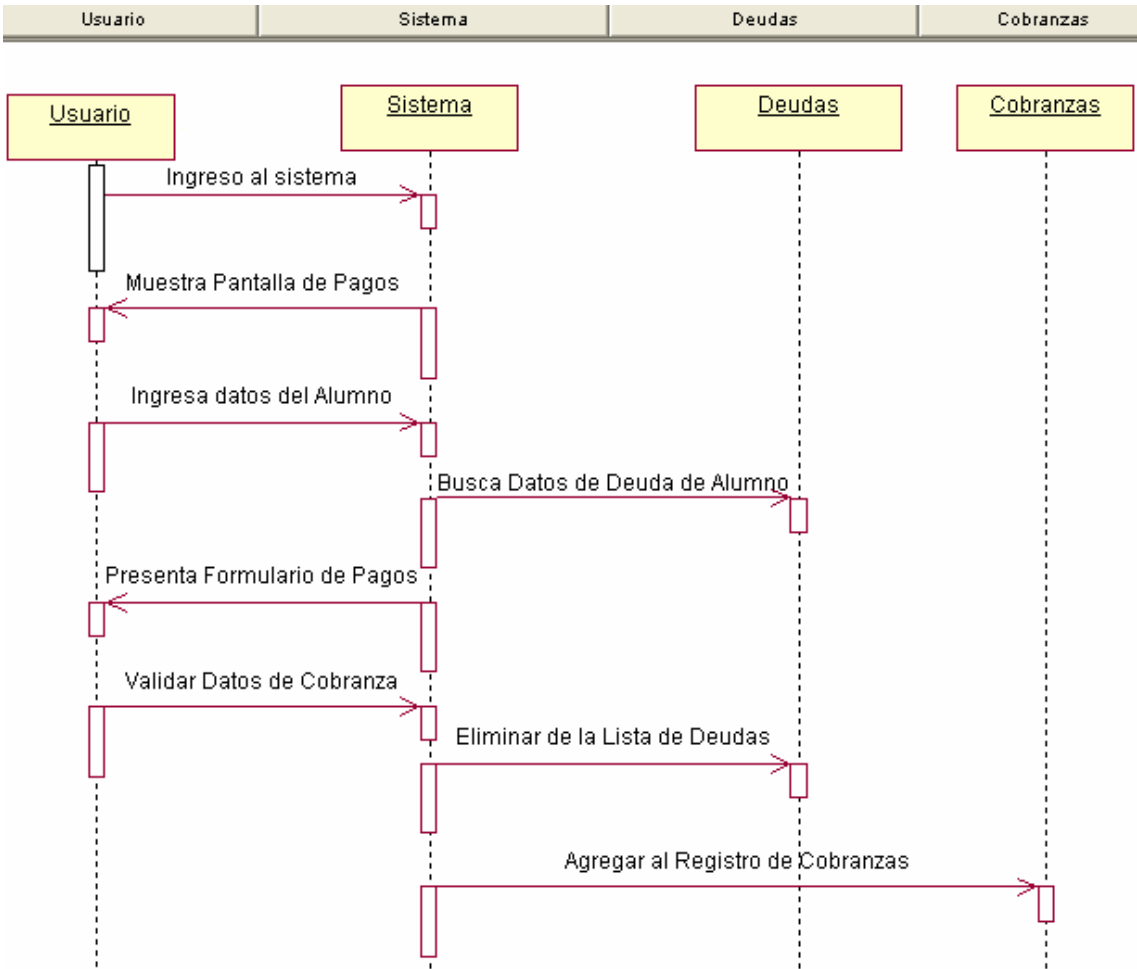


Figura 4.28: Diagrama de Secuencias - Realizar Pago de Alumno

4.5.2.2 Realizar Pago Fraccionado de Alumnos

4.5.2.2.1 Diagrama de Estados

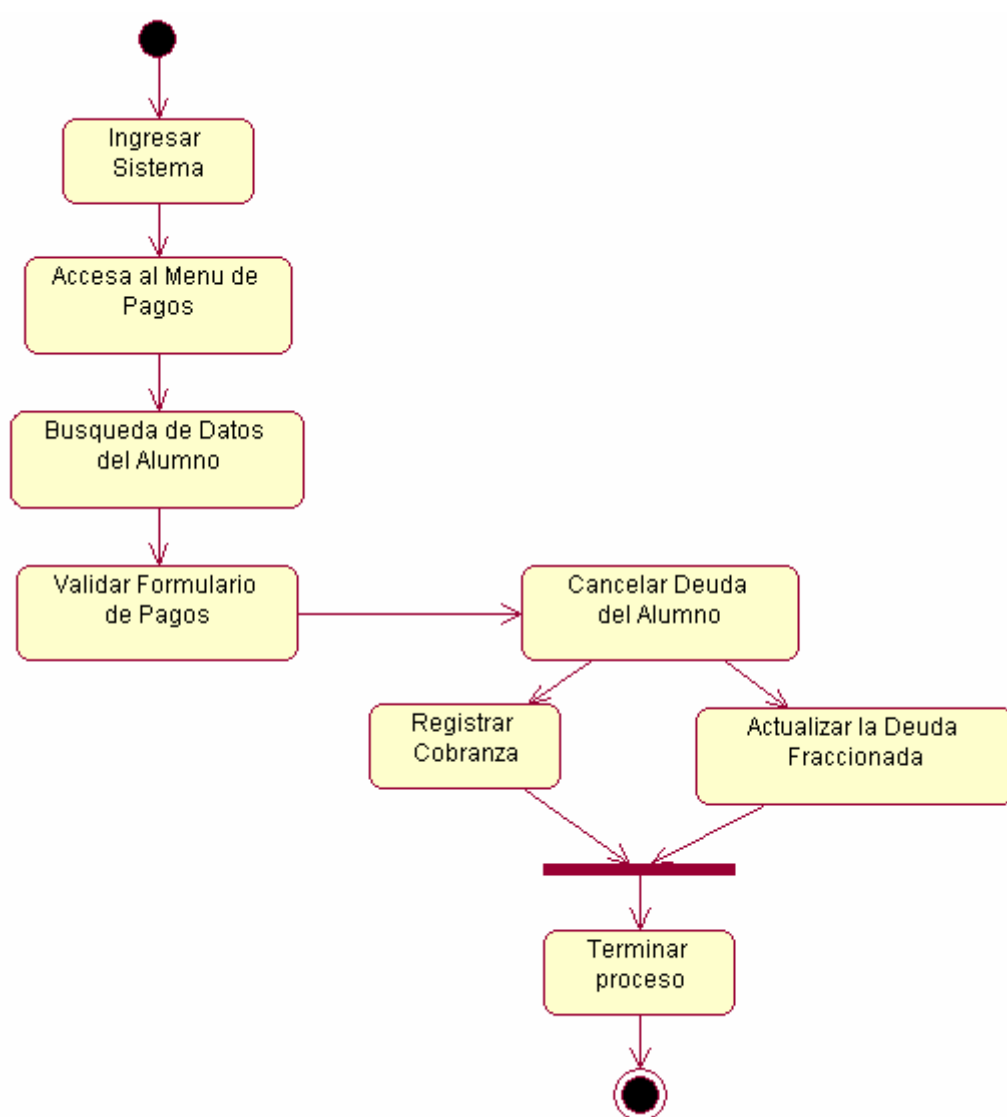


Figura 4.29: Diagrama de Estados - Realizar Pago Fraccionado de Alumnos

4.5.2.2.2 Diagrama de Secuencias

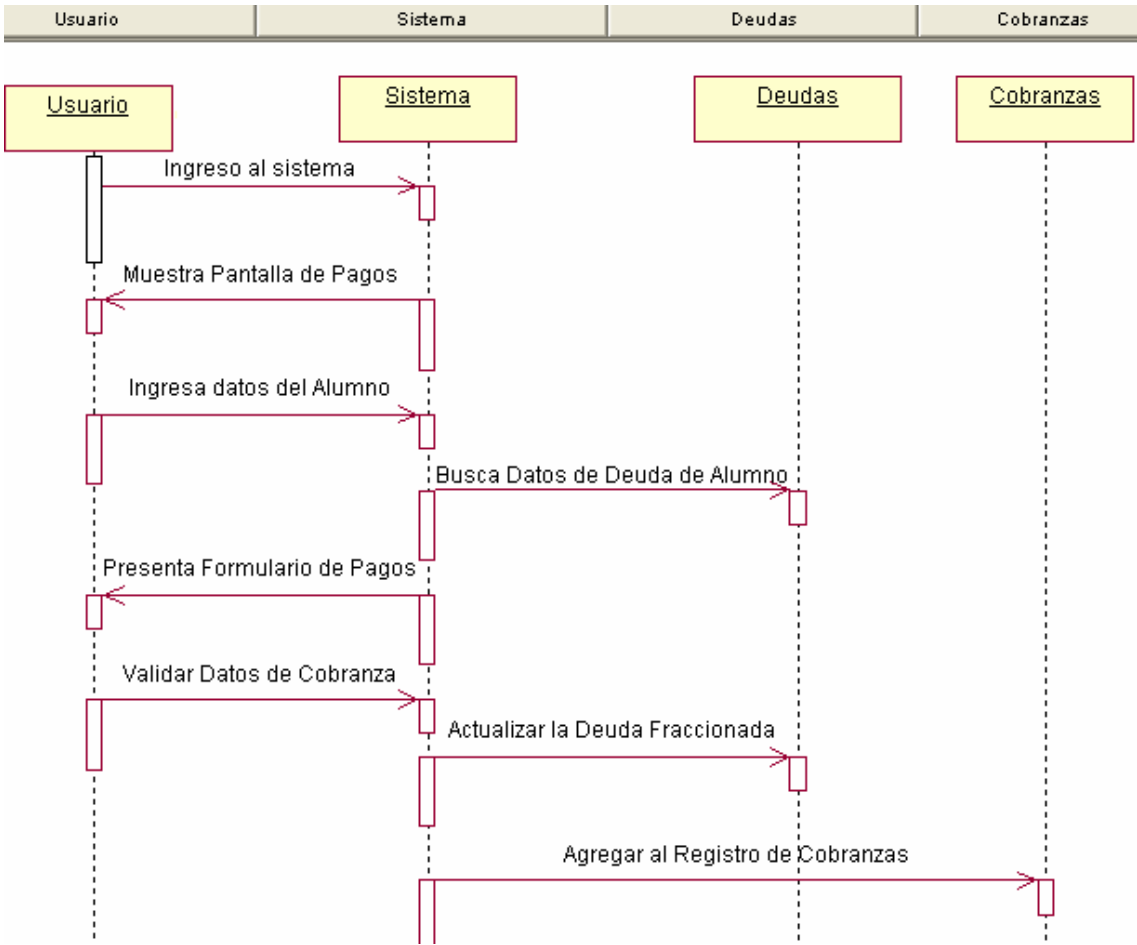


Figura 4.30: Diagrama de Secuencias - Realizar Pago Fraccionado de Alumnos

4.5.3 DISEÑO DE LA BASE DE DATOS

Para el diseño de la BD del Sistema de Pagos y Gestión de Cobranzas vía web se utilizó la herramienta DBDesigner 4, nos permitió el modelamiento de los datos aquí se presenta el Diseño Físico de la BD.

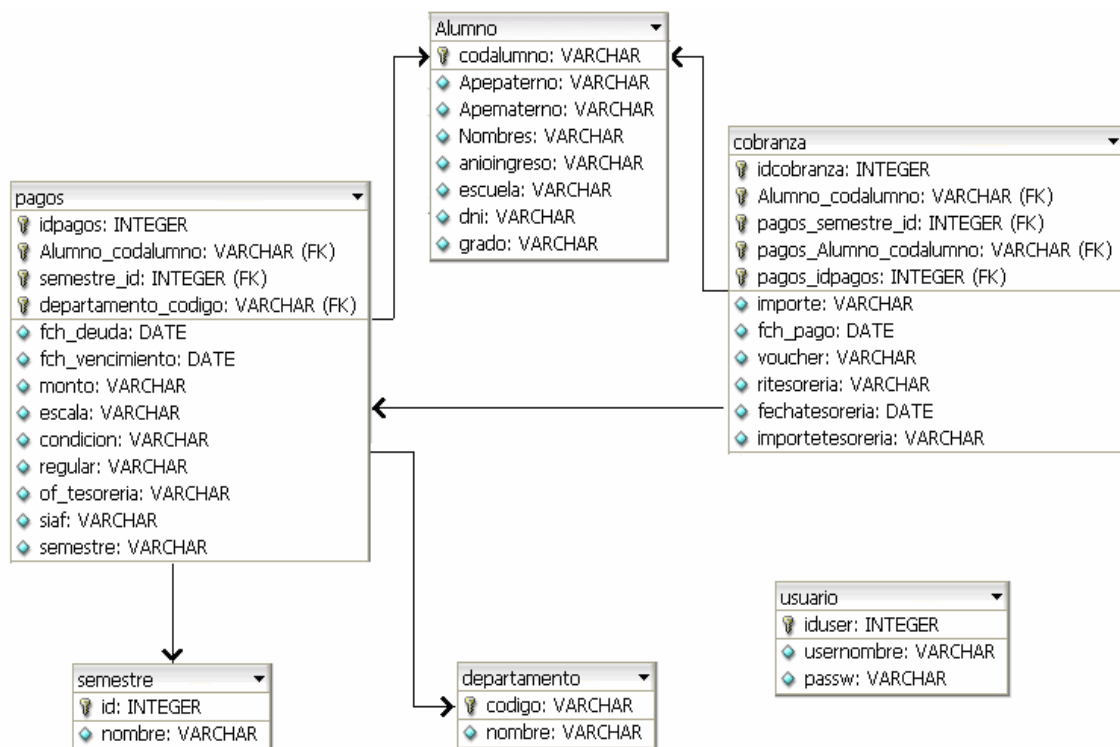


Figura 4.31: Diseño Físico - Diagrama Entidad Relación Elaboración Propia

Pagos: Aquí podemos guardar los datos de las deudas de los alumnos.

Cobranzas: Aquí podemos guardar los datos de las cobranzas registradas de los alumnos.

Alumno: Aquí podemos guardar los datos del alumno.

Usuarios: Aquí podemos guardar los datos de los usuarios que tendrán acceso al sistema.

Departamento: Aquí podemos guardar los datos del departamento de la deuda.

Semestre: Aquí podemos almacenar los semestres de los años.

4.5.4 CONFIGURACIÓN DEL SISTEMA

Tenemos pasos importantes para implementar como la instalación del driver de conexión como el visualizador de la Base de Datos.

4.5.4.1 Instalación de la conexión ODBC

Conectamos el MySQL ODBC 3.51 desde las herramientas administrativas del Panel de Control.

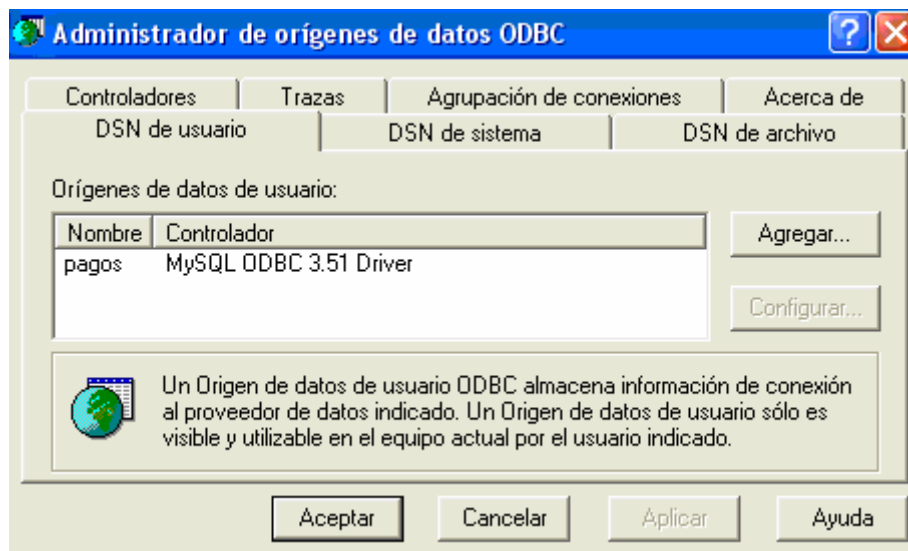


Figura 4.32: Configuración del odbc

4.5.4.2 Instalar servidor independiente de BD

Instalamos el WampServer2.0i luego ejecutamos phpMyAdmin y configuramos como en la figura.

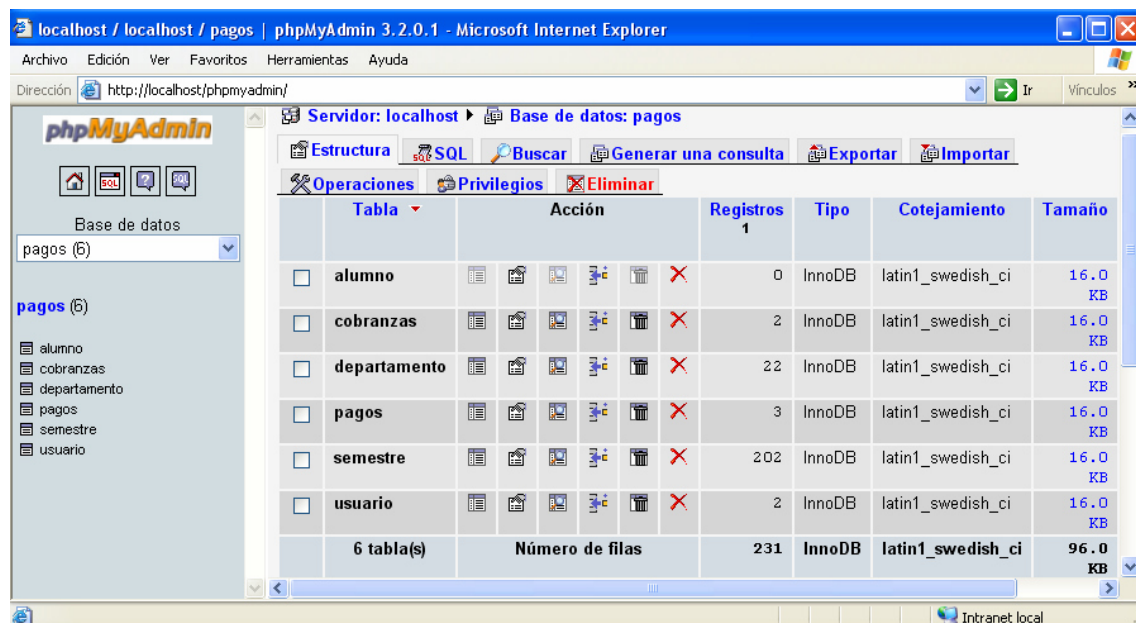


Figura 4.33: Configuración base datos

4.5.5 PROTOTIPO DEL SISTEMA

Se muestra el Sistema de Pagos y Gestión de Cobranzas de la FISI de UNMSM.

4.5.5.1 Ingreso de Login

El usuario ingresará el login y su password de acceso para este caso los usuarios son los administrativos de la FISI de la UNMSM.



Figura 4.34: Ingreso de Login

4.5.5.2 Menú Principal

Se presenta el menú principal con las opciones del sistema.



Figura 4.35: Menú Principal

4.5.5.3 Ingresar Deudas de Alumnos

El formulario para ingresar deudas de alumnos y sus datos referentes a este proceso como se muestra en la figura.

The screenshot shows a web browser window with the title 'Sistema de Pagos y Gestión de Cobranzas Vía Web - FISI - UNMSM - Microsoft Internet Explorer'. The address bar shows 'http://localhost:2151/Willy/Willy/Insertardeuda.aspx'. The main content area is titled 'Ingresar Alumnos con deudas' and contains a form with the following fields:

Codigo Alumno:	<input type="text"/>
Apellidos y Nombres :	<input type="text"/>
Departamento	Seleccione <input type="button" value="v"/>
Semestre de la deuda	2000-I <input type="button" value="v"/>
Fecha Inicio Deuda: (dd/mm/aaaa)	<input type="text"/> <input type="button" value="Calendar"/>
Fecha de Vencimiento (dd/mm/aaaa)	<input type="text"/> <input type="button" value="Calendar"/>
Total Deuda en Soles:	<input type="text"/>
Escala:	<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E
Condición:	<input type="radio"/> SIN COMPROMISO <input type="radio"/> CON COMPROMISO
Regular:	<input type="radio"/> REGULAR <input type="radio"/> NO REGULAR
Nro oficio a tesoreria:	<input type="text"/>
Co :	<input type="text"/>
Nro Oficio Siaf	<input type="text"/>
<input type="button" value="Grabar"/> <input type="button" value="Reporte de deudas"/> <input type="button" value="Ir al Menú"/>	

The status bar at the bottom shows 'Listo' on the left and 'Intranet local' on the right.

Figura 4.36: Ingresar Deudas de Alumnos

4.5.5.4 Cobranza del Pago del Alumno

El formulario para la cobranza del pago de alumnos y sus datos referentes a este proceso como se muestra en la figura.

The screenshot shows a web browser window with the title 'Sistema de Pagos y Gestión de Cobranzas Vía Web - FISI - UNMSM - Microsoft Internet Explorer'. The address bar shows 'http://localhost:2151/Willy/Willy/RealizarCobranza.aspx'. The main content area is titled 'Ingreso de Cobranza de Alumnos' and contains a form with the following fields and buttons:

Codigo Alumno:	<input type="text"/>	Buscar
Apellidos y Nombres :	<input type="text"/>	
Total Deuda (S/.) :	<input type="text"/>	
Fecha de Pago: (dd/mm/aaaa)	<input type="text"/>	
Total a Pagar (S/.)	<input type="text"/>	
Número de Voucher:	<input type="text"/>	
RI tesoreria:	<input type="text"/>	
Fecha Tesoreria: (dd/mm/aaaa)	<input type="text"/>	
Importe Tesoreria:	<input type="text"/>	
<input type="button" value="GRABAR COBRANZA"/> <input type="button" value="Cancelar"/> <input type="button" value="Ir al Menú"/>		

Figura 4.37: Cobranza del Pago del Alumno

4.5.5.5 Modificación de la Deuda del Alumno

El formulario para la modificación de la información de las deudas de alumnos y sus datos referentes a este proceso como se muestra en la figura.

Sistema de Pagos y Gestión de Cobranzas Vía Web - FISI - UNMSM - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost:2151/Willy/Willy/anadirdeuda.aspx> Ir Vínculos

Modificación Pagos de Alumnos

Codigo Alumno:	<input type="text"/>	Buscar
Apellidos y Nombres :	<input type="text"/>	
Departamento	Seleccione ▼	
Semestre de la deuda	2000-I ▼	
Fecha Inicio Deuda: (dd/mm/aaaa)	<input type="text"/>	
Fecha de Vencimiento (dd/mm/aaaa)	<input type="text"/>	
Importe en Soles:	<input type="text"/>	
<input type="checkbox"/> Incrementar deuda	<input type="text"/>	Agregar Cancelar
Escala:	<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E	
Condición:	<input type="radio"/> SIN COMPROMISO <input type="radio"/> CON COMPROMISO	
Regular:	<input type="radio"/> REGULAR <input type="radio"/> NO REGULAR	
Nro oficio a tesoreria:	<input type="text"/>	
Nro Oficio Siaf	<input type="text"/>	
Co :	<input type="text"/>	
Grabar Reporte de deudas Ir al Menú		

Listo Intranet local

Figura 4.38: Modificación de la Deuda del Alumno

4.5.5.6 Pago Fraccionado de las Deudas del Alumno

El formulario para el Pago Fraccionado de las deudas de alumnos y sus datos referentes a este proceso como se muestra en la figura.

Sistema de Pagos y Gestión de Cobranzas Vía Web - FISI - UNMSM - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost:2151/Willy/Willy/pagarfraccionado.aspx> Ir Vínculos

Ingreso de Pagos Fraccionado

Código Alumno	<input type="text"/>	Buscar
Apellidos y Nombres	<input type="text"/>	
Total Deuda (S/.)	<input type="text"/>	
Numero de Voucher:	<input type="text"/>	
FRACCION DE PAGO S/.	<input type="text"/>	
Fecha de Pago : (dd/mm/aaaa)	<input type="text"/>	
RI tesoreria	<input type="text"/>	
Fecha Tesoreria : (dd/mm/aaaa)	<input type="text"/>	
Importe Tesoreria	<input type="text"/>	

PAGAR FRACCIONADO Cancelar Ir al Menú

Listo Intranet local

Figura 4.39: Cobranza del Pago Fraccionado

4.5.5.7 Reporte de Alumnos con Deudas

El Reporte de Alumnos con deudas de los alumnos y muestra sus datos de este proceso como se presenta en la figura.

Departamento	Semestre	Codigo	Alumno	Fecha de Inicio	Vencimiento	Deuda (S/.)
RECTIFICACION DE MATRICULA	2009-II	07011318	Tavara Taboada, Julian Ricardo	15/10/2009	25/02/2010	36
REPITENCIAS	2009-II	08112311	Rodrigo Duran, Juan Alberto	15/12/2009	26/02/2010	21

Figura 4.40: Reporte de Alumnos con Deudas

4.5.5.8 Reporte de Cobranzas

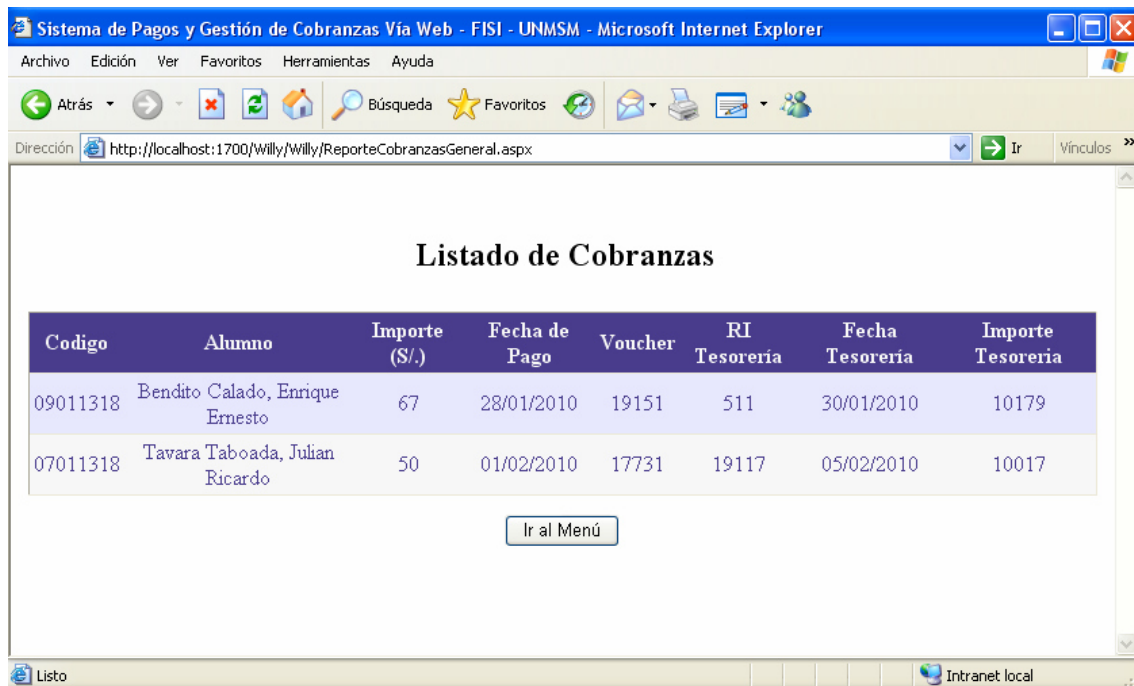
El Reporte de Cobranzas de los alumnos y muestra sus datos de este proceso como se presenta en la figura.

Codigo	Alumno	Acumulado (S/.)	Ultima Fecha de Pago	Ultimo Voucher
07011318	Tavara Taboada, Julian Ricardo	50	01/02/2010	17731
09011318	Bendito Calado, Enrique Ernesto	67	28/01/2010	19151

Figura 4.41: Reporte de Cobranzas

4.5.5.9 Listado de Cobranzas General

El Listado de Cobranzas General de los alumnos muestra los datos de este proceso como se muestra en la figura.



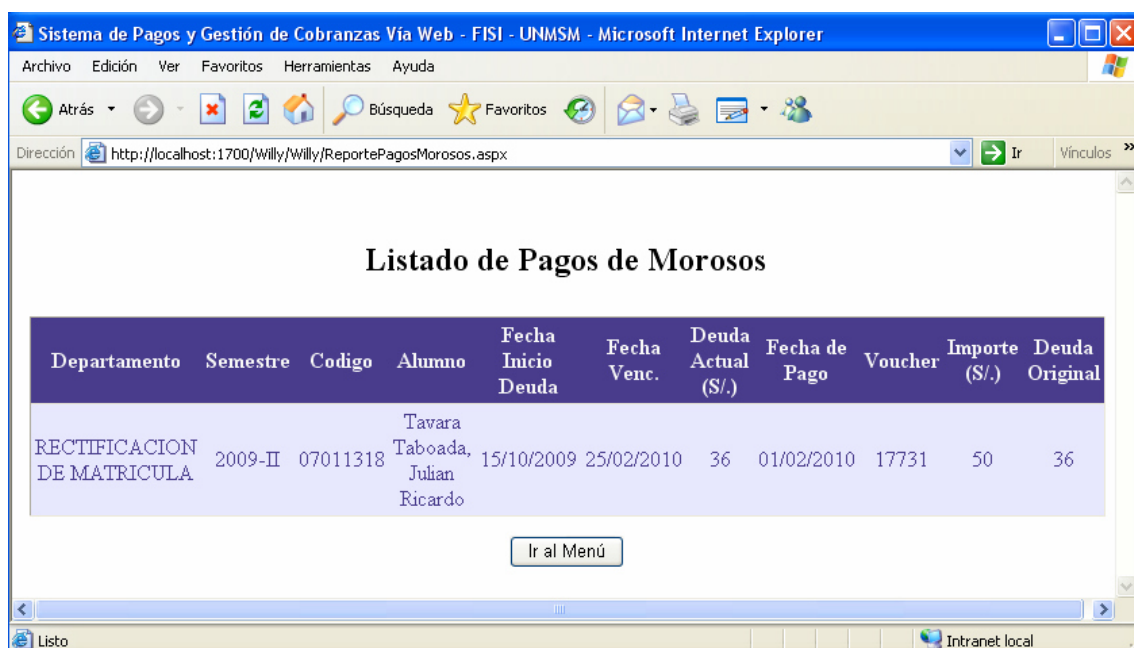
Codigo	Alumno	Importe (S/.)	Fecha de Pago	Voucher	RI Tesoreria	Fecha Tesoreria	Importe Tesoreria
09011318	Bendito Calado, Enrique Ernesto	67	28/01/2010	19151	511	30/01/2010	10179
07011318	Tavara Taboada, Julian Ricardo	50	01/02/2010	17731	19117	05/02/2010	10017

[Ir al Menú](#)

Figura 4.42: Listado General de Cobranzas

4.5.5.10 Reporte de Pagos de Morosos

El Reporte de Pagos de Morosos muestra los datos de los alumnos morosos como se muestra en la figura.



Departamento	Semestre	Codigo	Alumno	Fecha Inicio Deuda	Fecha Venc.	Deuda Actual (S/.)	Fecha de Pago	Voucher	Importe (S/.)	Deuda Original
RECTIFICACION DE MATRICULA	2009-II	07011318	Tavara Taboada, Julian Ricardo	15/10/2009	25/02/2010	36	01/02/2010	17731	50	36

[Ir al Menú](#)

Figura 4.43: Reporte de Pagos de Morosos

4.5.5.11 Listado de Deudas

El Listado de Alumnos con Deudas de los alumnos mostrando sus datos generales de este proceso como se presenta en la figura.

Departamento	Semestre	Codigo	Alumno	Fecha de deuda	Vencimiento	Deuda (S/.)	Escala	Condicion	Regular	Oficio	CO	siaf
RECTIFICACION DE MATRICULA	2009-II	07011318	Tavara Taboada, Julian Ricardo	15/10/2009	25/02/2010	36	E	SC	R	of 1031		siaf 101
REPTIENCIAS	2009-II	08112311	Rodrigo Duran, Juan Alberto	15/12/2009	26/02/2010	21	E	SC	R	of 1751		siaf 167

Figura 4.44: Listado de Deudas

4.5.5.12 Reporte de Deudas Acumuladas

Reporte de Deudas Acumuladas de Alumnos mostrando sus datos generales de este proceso como se presenta en la figura.

Codigo	Alumno	Escuela	Año Ingreso	Sexo	deudaoriginal	Deuda Acumulada (S/.)
07011318	Tavara Taboada, Julian Ricardo	Ing. sistemas	2007-I	M	36	36
08112311	Rodrigo Duran, Juan Alberto	Ing. sistemas	2008-I	M	21	21

Figura 4.45: Reporte de Deudas Acumuladas

4.5.5.13 Administración de Usuarios

Se presenta el menú de Administración de Usuarios con las opciones de Añadir, Modificar, Eliminar y Lista de Usuarios.

Usuarios			
Usuario	Password		
Admin	Admin	Editar	Eliminar
willy	willy	Editar	Eliminar

Nuevo Usuario	
Usuario	<input type="text"/>
Password	<input type="text"/>
Confirmar Password	<input type="text"/>
Insertar Cancelar	

[Retornar](#)

Figura 4.46: Administración de Usuarios

Para Ingresar Nuevos Usuarios

El formulario para Nuevo Usuario e ingresar el nombre de usuario y password como se muestra en el link Insertar.

Para Modificar datos de Usuarios

En el formulario para modificar datos de usuarios se muestra en el link Editar.

Para Eliminar Usuarios

El formulario para Eliminar usuarios de la BD se muestra en el link Eliminar.

4.5.5.14 Administración de Alumnos

Se presenta el menú de Administración de Usuarios con las opciones de Añadir, Modificar, Eliminar y Lista de Alumnos.

Alumnos

Codigo	Nombres	Direccion	Año Ingreso	Semestre Ingreso
--------	---------	-----------	-------------	------------------

[Editar](#) [Eliminar](#)

Nuevo Alumno

Codigo	<input type="text"/>
Nombres	<input type="text"/>
Direccion	<input type="text"/>
Año Ingreso	<input type="text"/>
Semestre Ingreso	<input type="text"/>

[Insertar](#) [Cancelar](#)

[Retornar](#)

Figura 4.47: Administración de Usuarios

Para Ingresar Nuevos Alumnos:

El formulario para Nuevo Alumno e ingresar los datos del Alumno como se muestra en el link Insertar.

Para Modificar datos de Alumnos

En el formulario para modificar datos de Alumnos se muestra en el link Editar.

Para Eliminar Alumnos

El formulario para Eliminar Alumnos de la BD se muestra en el link Eliminar.

4.5.5.15 Administración de Departamento

Se presenta el menú de Administración del Departamento con las opciones de Añadir, Modificar, Eliminar y Lista de Departamento.

The screenshot shows a web browser window with the title 'Departamento'. The address bar displays 'http://localhost:3260/Willy/Willy/Departamento.aspx'. The main content area features a table with two columns: 'Codigo' and 'Nombre'. To the right of the table are links for 'Editar' and 'Eliminar'. Below the table is a section titled 'Nuevo Departamento' containing two input fields for 'Codigo' and 'Nombre', each marked with a red asterisk. At the bottom of this section are links for 'Insertar' and 'Cancelar'. A 'Retornar' button is located at the bottom right of the page.

Figura 4.48: Administración de Departamento

Para Ingresar Nuevo Departamento

El formulario para Nuevo Departamento e ingresar su código y nombre como se muestra en el link Insertar.

Para Modificar datos de Departamento

En el formulario para modificar datos de Departamento se muestra en el link Editar.

Para Eliminar Departamento

El formulario para Eliminar Departamento de la BD se muestra en el link Eliminar.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Se desarrollo una propuesta para la construcción de un Sistema de Pago y Gestión de Cobranza Vía Web en la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, con un modelo propio y haciendo uso de las tecnología Web para el acceso desde internet.

Se registra, controla y emite los resultados de los pagos y cobranzas realizadas en un portal web, actualizando a nivel web la información más reciente.

Se recorta los plazos de pagos de los alumnos y sus procesos administrativos vinculados a matricula.

5.2 RECOMENDACIONES

Recordar que la construcción del sistema fue planteada para nuestra Facultad, de manera que los procesos internos de la FISI deben funcionar de manera integrada para mejorar el servicio a los alumnos.

El sistema estará sobre una plataforma orientada a objetos, esta visión de la metodología esto permite obtener un nivel conceptual de la aplicación en un primer plano, permitiendo la independencia de la plataforma de implementación y migración de la aplicación.

La implementación del sistema para la construcción y desarrollo se ha propuesto para un entorno por su alto grado de difusión y facilidad de mantenimiento y ampliación de funcionalidad, el utilizar las herramientas de software seleccionadas ha permitido desarrollar un software de calidad de buen rendimiento.

GLOSARIO DE TÉRMINOS

Análisis de sistemas: Estudio de una tarea o función para comprenderla y encontrar mejores maneras de realizarla.

Atributo: Representan los datos asociados al objeto, o lo que es lo mismo sus propiedades o características. Los atributos y sus valores en un momento dado, determinan el estado de un objeto.

Base de datos relacional: Colección de datos organizada y relacionada, para evitar duplicaciones y permitir la obtención de datos combinados, satisfaciendo la necesidad de usuarios con diferentes necesidades de información.

Cliente: Programa que demanda servicios de otra computadora llamada servidor, y se hace cargo de la interacción necesaria con el usuario.

Código fuente: Programa escrito en un lenguaje de programación de alto nivel por un programador.

Computadora: Es un sistema digital con tecnología microelectrónica capaz de procesar datos a partir de un grupo de instrucciones denominado programa. La estructura básica de una computadora incluye microprocesador (CPU), memoria y dispositivos de entrada/salida (E/S), junto a los buses que permiten la comunicación entre ellos.

Dato: Representación de un hecho o idea que puede ser manipulado y al cual se le puede asignar un significado.

Diagrama de flujo: Representación gráfica de los tipos y secuencia de operaciones de un programa o proceso.

Dominio: El nombre de dominio es un identificador único a través de la cual las computadoras se vinculan a Internet (por ej. para identificar sitios web y direcciones de correo electrónico). El sistema es jerárquico permitiendo la definición de subdominios de un dominio existente.

DNS (Domain Name System): Sistema que almacena y asocia diferentes tipos de información con nombres de dominio. Su función más importante es traducir el nombre de dominio a una dirección IP.

Framework: En programación orientada a objetos, diseño de subsistema generalizado para construir aplicaciones. Se compone de clases abstractas y de su colaboración objeto, así como de clases concretas.

Hardware: Los componentes físicos de la computadora, así como sus periféricos.

HTML (HyperText Markup Language): Lenguaje de composición de páginas de hipertexto para la WWW.

HTTP (HyperText Transfer Protocol). Protocolo de transferencia de hipertexto. Es el protocolo que permite navegar por la WWW

Información: Es el resultado del procesamiento de datos. Todo aquello que permite adquirir cualquier tipo de conocimientos.

Interfaz: Conexión entre dos componentes de hardware, entre dos aplicaciones o entre un usuario y una aplicación. También llamada por el término en inglés interface.

Internet: Red mundial de computadoras conectadas a través del protocolo TCP/IP. Es la más grande e importante red de redes interconectadas a través de routers.

Intranet: Denominación utilizada para referirse a la red interna de una empresa o institución.

ISP (Internet Service Provider): Proveedor de servicios de Internet.

LAN (Local Area Network): Red de área local. Es la forma en la cual se interconectan computadoras ubicadas en un mismo lugar a través de un cable de red.

Linux: Núcleo o kernel de sistema operativo compatible con UNIX®, que se puede utilizar en casi cualquier plataforma de computadora sin pagar costo de licencia y con libre acceso y modificación de su código fuente.

Login: Acción de conectarse a un sistema ingresando un nombre de usuario y una contraseña

Megabyte (MB): Medida de información equivalente a 1.024 kilobytes.

Memoria: Almacenamiento primario de una computadora, como la RAM o la ROM

Memoria principal: Lugar en el cual se almacenan datos e instrucciones en una computadora antes y durante su ejecución.

Memoria virtual: Una técnica de administración de memoria que permite utilizar un espacio del disco duro como si se tratase de memoria RAM.

Menú: Lista de opciones mostrada sobre una pantalla de las cuales el usuario puede seleccionar.

Nodo: Computadora o cualquier otro dispositivo conectado a una red

Online: Equipos o dispositivos que están en comunicación directa o encendidos.

Password: Contraseña utilizada para ingresar en una red o en un sistema de manera segura. Conjunto de caracteres alfanuméricos requeridos para acceder a una determinada red, sistema, aplicación o recurso.

Procesamiento de datos: Secuencia sistemática de operaciones realizadas sobre datos para obtener un resultado deseado.

Procesamiento en tiempo real: Técnica de procesamiento en que la actualización de los datos afectados por un evento se realiza a medida que sucede el evento causante.

Proceso: Manipular datos o realizar otras operaciones de acuerdo a un programa.

Programación: Creación de un programa informático, un conjunto concreto de instrucciones que una computadora u otro dispositivo informático puede ejecutar.

Programa: Secuencia de instrucciones que dirige a la computadora a realizar operaciones específicas para obtener un resultado deseado

Protocolo: Definición del sistema de comunicación de una computadora. Acuerdo entre diferentes sistemas para trabajar conjuntamente bajo un estándar común.

Red: Interconexión de una o más computadoras a través de hardware y software

Servidor: Computadora o programa que proporciona recursos y servicios a las computadoras conectadas a una red y al mismo tiempo gestiona el uso de esa red.

Sistema: Conjunto de elementos interrelacionados que trabajan juntos para obtener un resultado deseado.

Sistema operativo: Programa de control que dirige el hardware de una computadora. Por lo general es, en realidad, una colección de programas que interactúan juntos.

Software: Programas escritos en un lenguaje que la computadora entiende y puede ejecutar para realizar una tarea

TCP/IP: Conjunto de protocolos que rigen la transmisión de información en Internet.

Terminal: Dispositivo en un sistema o red de comunicación en el cual los datos pueden ingresarse o salir, pero no procesarse.

Testing: La prueba de un programa o un sistema para asegurar que funciona adecuadamente.

URL (Uniform Resource Locator). Localizador uniforme de recursos. Estándar que especifica un tipo de servicio en Internet, así como la localización exacta del archivo correspondiente. Dirección de un sitio web, universal y único a nivel mundial.

Usuario: Cualquier individuo que interactúa con la computadora a nivel de aplicación.

Windows: Denominación genérica de la gama de sistemas operativos de Microsoft® con prestaciones de GUI.

World Wide Web (www). Sistema de organización de la información de Internet a través de enlaces hipertexto. En sentido estricto es el conjunto de servidores que emplean el protocolo HTTP.

REFERENCIAS BIBLIOGRÁFICAS

BIBLIOGRAFÍA ESPECIALIZADA

LIBROS

- [JACOBSON,+,2000] JACOBSON, Ivar; BOOCH, Grady y RUMBAUGH, James, "El proceso unificado de desarrollo de software", Edit. Addison Wesley, 2000.
- [KRUCHTEN,2000] KRUCHTEN, P., The Rational Unified Process: An Introduction, Addison Wesley, 2000.
- [LARMAN, 2005] LARMAN, CRAIG, "Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development" 3rd edition, Prentice-Hall, 2005.

TESIS Y TESINAS

- [CARRILLO, 2007] CARRILLO LEIVA, GERMÁN LEIVA "Análisis y Diseño de un sistema de aulas virtuales de pregrado y postgrado en administración pública para el fortalecimiento institucional de gobiernos seccionales autónomos", Tesis para optar el título profesional de Ingeniero de Sistemas, Quito, 2007
- [LAGOS, 2008] LAGOS ROJAS, CÉSAR AUGUSTO, "Propuesta de implementación de un CRM para PYMES en el sector textil", Tesis para optar el título profesional de Ingeniero de Sistemas, Lima, 2008.

PÁGINAS WEB

- [ANAYA,2007] ANAYA VILLEGAS,ADRIAN, Programación Extrema
www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml
- [PALACIO,2007] PALACIO, JUAN, Flexibilidad con Scrum, 2000
www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf
- [PICON,2007] GUSTAVO PICÓN, Escalabilidad Web
<http://tabo.pe/talks/tabo-scalability.pdf>
- [QUINN,2002] QUINN, LIAM , Tips Sobre Accesibilidad
<http://htmlhelp.com/es/design/accessibility/tips.html>
- [RAWSTHORNE,2003] RAWSTHORNE, DAN, Comparing/Combining RUP, XP, and Scrum, www.netobjectives.com/files/events/download/rup_xp_scrum_pc_030326_ppt.pdf